View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Columbia University Academic Commons

Selected Legal Applications for Bayesian Methods

Edward K. Cheng

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2018

Copyright 2018
Edward K. Cheng
All rights reserved

ABSTRACT

Selected Legal Applications for Bayesian Methods

Edward K. Cheng

This dissertation offers three contexts in which Bayesian methods can address tricky problems in the legal system. Chapter 1 offers a method for attacking case publication bias, the possibility that certain legal outcomes may be more likely to be published or observed than others. It builds on ideas from multiple systems estimation (MSE), a technique traditionally used for estimating hidden populations, to detect and correct case publication bias. Chapter 2 proposes new methods for dividing attorneys' fees in complex litigation involving multiple firms. It investigates optimization and statistical approaches that use peer reports of each firm's relative contribution to estimate a "fair" or consensus division of the fees. The methods proposed have lower informational requirements than previous work and appear to be robust to collusive behavior by the firms. Chapter 3 introduces a statistical method for classifying legal cases by doctrinal area or subject matter. It proposes using a latent space approach based on case citations as an alternative to the traditional manual coding of cases, reducing subjectivity, arbitrariness, and confirmation bias in the classification process.

Contents

List of Tables

List of Figures

Acknowledgements

As the old expression goes, it takes a village, and nothing could be more true about finishing a dissertation.

My first note of thanks goes to David Madigan. He had the imagination and the courage to take on the crazy idea of a law professor who wanted also to become a statistician. My experience in this statistics program has fundamentally changed the course of my scholarly career and indeed the way I view the world. None of that would have happened without David, whose mentoring and encouragement always seemed to make the finish line seem possible.

I would like to thank present and past members of my dissertation committee, Liam Paninski, John Cunningham, Rahul Mazumder, Bert Huang, and Lauren Hannah, who have devoted time and effort to engage with my work. Thanks also to my co-authors on the Attorneys' Fees paper, Paul Edelman and Brian Fitzpatrick, as well as my research assistants on the Publication Bias paper, Maria Ramrath and Nicole Ballante.

I am grateful to comments that I received over the years on the three papers that make up this dissertation. For comments on Chapter 1, I thank Valena Beety, Mark Brandon, Jenny Carroll, Richard Delgado, Paul Edelman, Joe Fishman, Brian Fitzpatrick, Tracey George, John Gross, William Hubbard, Nancy King, Richard Leo, Aran McNerney, Alex Nunn, Mike Pardo, Jennifer Shinall, Ganesh Sitaraman, Fred Vars, Mike Vandenbergh, Albert Yoon, two anonymous reviewers, participants at an invited session of the Royal Statistical Society at the 2016 Joint Statistical

Meetings, participants at the 2017 AALS Annual Meeting, and workshops at the University of Alabama School of Law, Penn State Law, and Vanderbilt Law School. For comments on Chapter 2, I thank Ben Alarie, Albert Yoon, and participants at Law & Economics Colloquia at the University of Toronto and George Mason University. And for comments on Chapter 3, I thank Paul Edelman, Tracey George, Chris Serkin, and Suzanna Sherry. Thanks also to Charles McNamara for sharing his LaTeX dissertation template, Alex Bergier and Columbia University's High Performance Computing (HPC) cluster for technical support,and Dean Chris Guthrie and Vanderbilt Law School for institutional support.

As this dissertation marks the end of a very long educational journey, I would also like to take a moment to acknowledge all of my teachers who have given selflessly to me over the last four decades, and who in many ways have made me the scholar I am today. I would like especially to acknowledge: the late David Wagner, my high school Model United Nations advisor and history teacher, who inspired me to become a lawyer; Mort Horwitz, my Torts professor, who inspired and then guided me to become a professor; and finally Margaret Berger of blessed memory, my much-missed colleague and mentor in the world of evidence law.

Last but not least, my heartfelt thanks to my family. Thanks to my wife Jenny, who helped me dream up this project and then tirelessly and unflinchingly guided and supported me through it. She is a true woman of valor. Thanks to my parents, Stanley and Emy Cheng, for supporting me throughout my education and for encouraging me to do this Ph.D. seemingly forever. They instilled in me the focus and the work ethic that made all this possible. And thanks to my sons, Isaac, Aaron, and Eli, my brother Nelson, and my in-laws Peter and Ethel Diamond.

EKC

Nashville, TN

To my family

Introduction

Two decades before R.A. Fisher began his pathbreaking statistical work at Rothamsted Experimental Station, future United States Supreme Court Justice Oliver Wendell Holmes had already predicted that "for the rational study of the law the black-letter man may be the man of the present, but the man of the future is the man of statistics and the master of economics." (Holmes, 1897). On the economics score, the rise (and in some spheres, the dominance) of "law and economics" in modern legal thought has proven Holmes prescient. And while Holmes's prediction about law and statistics has not yet come entirely to pass, recent trends toward empirical legal scholarship, electronic discovery, and the use of machine learning in legal contexts suggest that we may be on the cusp of his envisioned world.

This dissertation offers some exciting ways in which statistics can help solve legal problems. It examines three difficult law-related problems — the assessment of judicial behavior, the division of attorneys' fees, and the classification of cases — and asks how modern statistical tools can aid or replace traditional expert intuition and judgment in these contexts. Chapter 1 addresses case publication bias. In trying to assess the current state of law, lawyers have typically relied on the available (published) case law. What happens if that case law exhibits publication bias, in which certain outcomes are more likely to be observed than others? Is there a way to de-bias our assessments? Chapter 2 tackles the division of attorneys' fees. Often at the end of mass litigation, courts must divide a pool of attorneys' fees among the participating firms. Traditionally, such division is done in ad hoc ways by the judge or a special master. Can statistical tools provide a better mechanism? Finally, Chapter 3 examines case classification. Historically, cases are classified manually and subjectively. Can statistics provide a more objective and automated classification method?

These three legal problems may seem entirely disparate, and indeed each of the the three chapters develops its own statistical models to address the specific problem posed. But taking a broader view suggests some deeper conceptual connections. All three solutions are (at least in part) solved using latent variable models. The case publication bias solution involves multiple systems estimation, in which a model infers the size of a hidden (unobserved) population based on the observed members. One of the fee division models infers the (hidden) just distribution of fees based on peer ratings. And finally, the case classification model uses case citations to infer a latent case space. That latent space enables us to locate a case's proximity to other cases.

That latent variable models have so much to offer the legal system is in many ways unsurprising. At the heart of most legal tasks is the need or desire to reveal some hidden truth. Latent space models are therefore a natural fit. What follows offers a glimpse of the power and versatility of latent space models in solving legal questions. Going forward, one suspects that we have only scratched the surface.

1 Detection and Correction of Case Publication Bias[1]

The observable case law in databases like Westlaw or Lexis is not a representative sample of the universe of legal decisions. To be captured in such a database, two things must happen. First, the court must decide to write an opinion, rather than rule orally or issue a summary order. Second, the database must somehow capture that opinion. For example, the court may choose to publish the opinion in an official reporter, which will make inclusion in a database highly likely. Or the court may leave the opinion "unpublished," in which case inclusion in the database will be less likely and depend on the specific database's mechanisms for collection and selection. If these paths to observation happen to be correlated with case outcome, then the observable case law will suffer from selection bias, and any inferences drawn about the law will be invalid. This possibility that certain legal outcomes may be more likely observed than others is what I term "case publication bias."

Case publication bias presents a serious problem for both legal practice and research. Like other hazards, it perhaps poses the greatest danger to those unaware of it. For example, legal academics, attorneys, and jurists often conduct informal reviews of the case law to detect trends in the law or to predict outcomes.[2] If the observable case law is biased, then those impressions and predictions will be distorted and misleading.

[1] This chapter is forthcoming in *Journal of Legal Studies* (2018).

[2] These informal reviews of course have many dangers, but are frequently done anyway. (Baude et al., 2017).

But even for those cognizant of the problem, case publication bias presents a formidable obstacle. Researchers doing systematic studies of the case law need some way to detect and remove the bias to produce valid studies. Conventionally, researchers have tried to find comprehensive data sources that exhaustively catalog all case outcomes, but these are often unavailable or limited in scope. Thus, even careful researchers are frequently forced to muddle through using the observed case law. The best they can do is acknowledge their limitations and make arguments as to why the bias may not be severe.

In this Chapter, I explore the problem of case publication bias, with a particular emphasis on evidence law. I argue that there are structural reasons to believe that the case publication bias problem is particularly acute in evidence (although similar conditions may pertain elsewhere). Consequently, we cannot merely muddle through using the observable case law on evidence; we need to address the bias directly. Then, I propose a new method for detecting and correcting case publication bias based on ideas from multiple systems estimation (MSE), a technique traditionally used for estimating hidden populations in fields like ecology, human rights, and public health. The goal of the chapter is thus not only to increase awareness of case publication bias, but also to provide the academy, bench, and bar with a tool to address it.

The Chapter proceeds as follows. Section 1.1 motivates the discussion by describing the reasons why case publication bias may be especially acute in the evidentiary context. Section 1.2 discusses methods for detecting and correcting publication bias. It surveys existing approaches, introduces the intuition behind multiple systems estimation, and then develops a series of detection models. Section 1.3 turns to applications. To validate the method, I first apply the proposed model to simulated datasets in which I control the level of publication bias present. I then apply the model to a newly compiled dataset of evidentiary rulings dealing with false confession expert testimony. Section 1.4 discusses the assumptions and limitations of the model and some of the policy implications arising out of the results.

1.1 The Problem

Case publication bias creates problems for both informal (non-systematic) reviews of case law as well as more sophisticated analyses. For better or worse, informal reviews of existing case law are practically ubiquitous in the legal world. (Baude et al., 2017). Associates produce case law surveys to provide background to a legal team and support for motions or briefs. Traditional doctrinal treatises and law review articles use case law databases to detect trends or ascertain the state of the law. Courts in turn rely on and are influenced by these works or their own reviews, especially when they have discretion or lack expertise, or when the state of the law is uncertain.

For example, in the wake of Crawford v. Washington (2004), when the "testimonial" status of 911 calls was unclear, courts such as the Sixth Circuit in United States v. Hadley (2005) cataloged the published decisions of other courts in detail before making their own decision. (See also United States v. Brito, 2005). With respect to false confession expert testimony, courts have similarly relied on such reviews to determine admissibility as well as to assess ineffective assistance of counsel claims.

(Vent v. State, 2003; State v. Buechler, 1998).[3] To be sure, courts do not treat these informal reviews as dispositive or as a mere head-counting exercise, but the influence and persuasive power of them is unavoidable. The influence exists whether the cases are selected for "official" publication or not, as well as whether they are classified as "precedential" or not, because the existing body of available case law frames the terms of the debate. Even if not explicitly acknowledged, they exert social pressure on judges. (See, e.g., Nolan et al., 2008, 913).

Left undetected, the consequences of case publication bias can be considerable for informal reviews. (Cf. Delgado and Stefancic, 1989). Biased conclusions may influence judges in making their decisions. Attorneys may erroneously conclude that their preferred theory has no chance of success, and thus encourage their client to plea or not take a case at all. Legal treatises may codify the biased picture of the case law and compound the problem by passing the misimpression along to readers.

More rigorous, systematic surveys of case law eliminate many of the weakness and pitfalls of informal reviews, but they must still contend with the case publication bias problem to reach valid conclusions.[4] One solution is to use comprehensive databases that include all unpublished cases (or even all filed cases), such as the well-known dataset from the Administrative Office of the United States Courts, which tracks all

[3]Indeed, in Lunbery v. Hornbreak (2010), a concurring judge argued that counsel's failure to offer false confession expert testimony was unreasonable given its admissibility *in other cases*. (Lunbery v. Hornbreak, 2010, 764). One curious if not Kafkaesque example is found in Brooks v. State (1999), in which the Mississippi Supreme Court relied on a set of cases – including a later proven wrongful conviction – to justify blanket admissibility for bitemark evidence. The defendant in *Brooks* was ultimately exonerated through DNA evidence. (Fabricant and Carrington, 2016, 8–9).

[4]For example, Fitzpatrick (2010b) criticizes previous studies of class action settlements for relying entirely on "district court opinions that were published in Westlaw or Lexis."(Fitzpatrick, 2010b, 829)

cases in the federal system. (Hubbard, 2013). The problem is that the AO database contains limited information, and depending on the research question, an appropriate comprehensive database may not be available. As the legal academy and the broader legal community shift toward more systematic reviews of case law, (Baude et al., 2017; Bar-Gill et al., 2017), the need to detect and address case publication bias will only grow.

1.1.1 Case Publication Bias in Evidence

In some contexts, the structure of legal decisionmaking makes concerns over case publication bias especially acute. Evidence is one of those contexts. Trial judges make many evidentiary decisions throughout the course of a trial, but efficiency and practicality militate that most of those decisions be oral or accompanied by limited written orders, making them unlikely to be included in databases. What then causes a trial judge to produce a written evidentiary opinion? Presumably, there are many factors, including the novelty of the issue and the judge's intellectual interests, but one situation likely to result in a written opinion is when a court excludes critical evidence offered by a criminal defendant. Because the evidence is critical, the ruling may be outcome determinative or nearly so. And because the court is excluding not admitting evidence, the ultimate responsibility for the outcome now rests with the judge, not the jury (as in the case of admissions). In these situations, a judge may therefore feel a particular obligation to produce an opinion in keeping with the rule of law, and as a consequence, legal reporting services may be more likely to capture high-stakes exclusion decisions, as opposed to admissions or non-critical exclusions.

Further, since appellate courts write and publish opinions more regularly than trial courts, the typical way to observe evidentiary decisions is through appellate decisions, not trial ones. But because appeals themselves are not representative of the underlying case population, we have yet another potential biasing mechanism. (Stith, 1990). For example, because of double jeopardy rules (and the rarity of interlocutory appeals), most criminal appeals arise from defendant convictions. Appeals of criminal convictions, however, will tend to feature trial court evidentiary decisions that were *adverse* to the defendant, including exclusions of the defendant's evidence. (Risinger, 2007, 468-69). Appellate opinions will thus tend to feature exclusions of the defendant's evidence rather than admissions of the defendant's evidence. And because evidentiary decisions are usually only reviewed deferentially on appeal, the appellate courts will tend to affirm those exclusions.[5]

False confession expert testimony (FCET) is a perfect storm of the biasing mechanisms just discussed. False confession experts, psychologists or sociologists who testify about the psychological factors that enhance the risk of a false confession, are often critical witnesses for criminal defendants. Confessions are highly damaging evidence against the defendant, and lay jurors are likely to find counterintuitive (if not implausible) the idea that someone would confess to a crime that he did not commit. Consider what happens with these admissibility decisions: If the court excludes a false confession expert, it may write an opinion explaining why it is excluding

[5]The deferentially standard of review necessarily means that affirmances carry less precedential weight, but recall that our focus is on the impression given by the observable case law and its psychological effect, rather than the strictly legal value.

critical evidence and effectively gutting the defendant's case. The opinion may then be selected for publication in an official reporter, or at minimum, will be captured by an electronic database. By contrast, if the court admits the expert, the witness becomes just another witness at trial, generating no special need to write an opinion. Databases are thus more likely to capture exclusions than admissions. A similar bias appears at the appellate level. If the trial court excludes the expert, conviction is likely – after all, the defendant confessed.[6] By contrast, if the court admits the expert, conviction is less likely, at least on the assumption that false confession experts are helpful to the defense. Thus, the pool of cases seen by the appellate court will be skewed toward exclusion, and given the deferential review standard, all of the appealed (and subsequently observed) cases are more likely to be affirmances of exclusions.

1.1.2 Case Publication Bias Generally

Publication bias writ large is of course well known in fields like medicine (Easterbrook et al., 1991), economics (Roberts and Stanley, 2006), and psychology (Rosenthal, 1979). The problem of case publication bias in the legal world, however, has received comparatively less attention, though a number of scholars have noted its presence and expressed concerns. (Hubbard, 2017; Fitzpatrick, 2010b). There have also been a few prior studies specifically attempting to detect and quantify the severity of the

[6]An alternative possibility is that a plea bargain will occur after the evidentiary determination, making the case disappear from the observable record. However, given these are false confession cases, one might expect innocent defendants to press on with the litigation. There is no particular reason to believe that the availability of pleas will skew the pool in one direction or the other.

bias. Siegelman and Donohue (1990) showed that the rate at which employment discrimination cases appeared in the published case law was dependent on case outcome. To measure the level of bias, it compared the prevalence of case outcomes in the Administrative Office of the United States Courts (AO) dataset with the published case law. (Siegelman and Donohue, 1990). Lizotte (2007) reported similar bias among summary judgment motions. It found that "[s]ummary judgments awarded to plaintiffs were more likely to appear online than were judgments awarded to defendants, and appealed judgments were more likely to be available that those that were not appealed."(Lizotte, 2007). Merritt and Brudney (2001) considered factors that lead to appellate opinions being published in official case reporters, though it did not test specifically for the influence of case outcome.[7]

Most recently, Wininger and Cecil (2015) explored differences among the sources of data commonly used for empirical legal studies, expressing concerns about possible biased sampling. Wininger and Cecil (2015) collected 12(b)(6) dismissal orders from three electronic sources: Westlaw, PACER, and the federal judiciary's Case Management / Electronic Case Files (CM/ECF) system. The authors found that while none of the databases were perfect, some were better at capturing certain courts or types of cases than others. Notably, the study did not find a significant difference in legal outcomes between the Westlaw and PACER databases. (Wininger and Cecil, 2015, 17).

[7]The article does report that certain circuits "encourage[] or require[] publication of decisions reversing the lower court or agency," though the article's definition of publication is restricted to appearance in official reporters.(Merritt and Brudney, 2001, 77, 81)

1.2 Methods

1.2.1 Bias Detection Methods

As previously discussed, the primary way to detect and address case publication bias thus far has been through the use of a comprehensive comparison dataset, such as the one provided by the AO. Finding a comprehensive comparison dataset, however, is difficult in many contexts. For example, there is no obvious way to construct such a dataset for studying evidentiary rulings. The AO neither codes for evidentiary determinations (because they are not the "topic" of a case or the substantive claim), nor does it cover state courts. At the same time, manual compilation of a complete dataset on evidentiary rulings is infeasible. The vast majority of evidentiary rulings are made orally and recorded only in trial transcripts, which are often available only in paper form, geographically dispersed, and expensive to obtain.[8] Worse yet, specific evidentiary contexts, such as rulings on false confession expert testimony, are thinly distributed among the population of transcripts, rendering traditional sampling techniques ineffective. This thin distribution makes direct surveys of judges infeasible as well, since most judges will have had no exposure to FCET and will be reluctant to comment on matters that may arise in the future.

Standard approaches to publication bias from the scientific and social scientific literatures are also inapplicable to case publication bias. Publication bias in scientific journals arises from p-value bias, the desire of journals to publish statistically

[8] In many jurisdictions, trial transcripts are the property of the court reporter, who sells copies to interested parties.

significant results. The approaches for detecting and correcting p-value publication bias accordingly take advantage of the underlying statistical structure. For example, one conventional method for detecting publication bias, the funnel plot, plots the reported effect sizes in the published studies against the estimated variances. (Duval and Tweedie, 2000; Givens et al., 1997). Asymmetries in the plots then become the basis for detecting and correcting publication bias.[9] Another method proposed by Simonsohn et al. (2014) uses the expected distribution of p-values among a group of studies to do correction and detection. Case publication bias lacks these underlying quantitative structures, preventing the use of these techniques.

1.2.2 Multiple Systems Estimation

To remedy some of the limitations discussed above, this Chapter proposes tackling the legal publication bias problem using techniques from multiple systems estimation (MSE). Multiple systems estimation has its conceptual origin in capture-recapture methods used in ecological studies. (Amstrup et al., 2005). The intuition behind MSE is readily shown through an ecology example: Suppose a researcher wishes to determine the number of fish in a pond. On Day 1, the researcher catches, tags, and releases ten fish from the pond. On Day 2, the researcher catches another ten fish, and from this, she is able to estimate the pond population. If the researcher catches

[9] Across multiple studies, we expect the reported effect sizes to be symmetrically distributed around the "true" effect size. The tightness of that distribution will be a function of the estimated variance – for example, large samples sizes will have tighter distributions. When we plot effect size against the reciprocal of the estimated variance for all available studies, we expect a funnel-shaped scatterplot that is wide for high-variance studies and narrow for low-variance studies. Asymmetries in the funnel plot imply publication bias, because the "missing" points are instances in which journals chose not to publish negative results. One can then "complete" of fill-in the funnel to correct the publication bias. (Duval and Tweedie, 2000)

only tagged fish, then the population is likely close to ten. Since we assume the researcher catches fish at random, it is highly unlikely that she will keep catching the same ten fish unless those are the only fish available for catching. If the researcher catches no tagged fish, then the population is at least twenty, but likely far larger, since the chance of catching at least one tagged fish is fairly high unless there are many other individuals to select from.

Conceptually, the capture-recapture method estimates population size using overlaps between different lists. The fish from Day 1 represent one list, or one sample from the population. The fish from Day 2 represent a second list or sample. The tagging is merely the means by which we keep track of the overlap between the two lists. Heavy overlap suggests a small underlying population, whereas light overlap suggests a large underlying population.

Capture-recapture implicitly makes several, strong assumptions. For example, all fish and all lists have the same probability of capture – i.e., no subgroup of fish is particularly gullible, and the Day 1 and Day 2 catches are of equal number. MSE provides the means for relaxing these assumptions. It uses regression modeling to control for differences in capture probabilities, enabling researchers to use existing lists, rather than collecting independent samples as in the fish pond example. Among other things, researchers have used MSE to estimate the number of deaths or human rights violations in conflict areas, (Lum et al., 2013; Seybolt et al., 2013; Fienberg and Manrique-Vallier, 2009; Asher et al., 2008), to estimate populations for public health research, (Laska, 2002; Madigan and York, 1997; Madigan et al., 1995; Regal and Hook, 1984), and to make corrections to the census, (Brown et al., 1999). In all of these contexts, the underlying population is hidden, but researchers use existing lists to estimate the full population size. For human rights violations, for instance, researchers use lists of victims compiled by various governmental and non-governmental organizations. (Lum et al., 2013). For the census, the "capture" is the census itself, whereas the "recapture" is a post-enumeration survey. (Brown et al., 1999).

The case publication bias problem is similar in structure. Although lawyers often think of the "case law" in monolithic terms, the observed case law is actually the aggregate of many legal research databases. Some databases, like Westlaw and Lexis, are more comprehensive than others and thus have higher capture probabilities. Some cases — whether due to the court, subject matter, or other factors — are more likely to be observed than others. Any workable model must thus account for these variations. In any event, the various databases overlap imperfectly, and these overlaps and discrepancies are what will permit us to draw inferences about the cases that are otherwise unobserved.

The existing MSE literature offers a variety of statistical approaches for estimating population size. (Bishop et al., 1975; Madigan and York, 1997; Fienberg et al., 1999; King et al., 2008). The publication bias problem, however, is not a population size problem and therefore requires an extension of the MSE approaches. Below, I extend the Rasch approach from Fienberg et al. (1999) for use as a method of detecting publication bias. (See also Pelle et al., 2016; Bartolucci and Forcina, 2001) I will also show how the model estimates can be used to correct for the detected bias so one can estimate the true underlying population characteristic, which in this case is admissibility rate.

1.2.3 Independence Model

We start with lists of relevant cases from the available legal research databases. Suppose we have K unique cases and L lists or databases. Our dataset then consists of a $K \times L$ matrix in which the entries (Y_{kl}) represent whether case k was observed (or "published") in a list l, as seen in Figure 1.1. We also collect information on the ruling in case k, i.e., whether the trial court admitted or excluded the evidence, which we label A_k.

	$List_1$	$List_2$	$List_3$	$List_4$	$\ldots$	$List_L$
$Case_1$	1	0	0	0	$\ldots$	1
$Case_2$	1	0	1	1	$\ldots$	1
$Case_3$	0	1	1	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$Case_K$	0	1	1	0	$\ldots$	0

Table 1.1: Case-List Data for Model (1 = case observed on list, 0 = case not observed on list)

Given this data, we can construct a probability model for when case k is observed in list l. For this first model, we assume that the various case databases are independent. The probability of observing case k in list l then depends on a variety of unknown (latent) attributes. As previously mentioned, certain cases will attract greater attention, either because the opinion is better written, the facts are more interesting, or the case has a higher profile. Similarly, some lists will be more comprehensive than others – major legal search databases such as Westlaw and Lexis will have a greater probability of observing a given case than smaller databases. With this backdrop, we propose the following regression model:

$$Y_{kl} \sim \text{Bernoulli}(\phi_{kl})$$

$$\text{logit}(\phi_{kl}) = \beta A_k + \theta_k + \gamma_l \tag{1.1}$$

In this model, ϕ_{kl} is the (latent) probability that case k will be observed in list l. We do not observe ϕ_{kl}, but instead only see the observation outcome, which is Y_{kl}. β is the parameter of interest, for it captures how much more likely an evidentiary admission is to be observed (on any list) than an evidentiary exclusion. Under our working theory of publication bias in the criminal context, we would expect β to be negative since A_k is the indicator variable for admission. The other parameters are random effects that control for the latent attributes previously discussed: θ_k accounts for case variations, and γ_l controls for list variations. We can reasonably model the noteworthiness of cases as being normally distributed, so that the θ_k arise from a common normal distribution.[10] We do not do so for the lists, however, because we anticipate them to be bimodal. Relatively speaking, the major databases (Westlaw and Lexis) are extremely comprehensive, while minor lists (like a BNA reporter or a website) may be more hit-or-miss. We therefore give γ_l and the other parameters flat normal priors, except for the hyperparameters for variance, which are given weak Cauchy priors as suggested by Gelman (2006).

The parameters in Eq. 1.1 can then be estimated using standard Markov Chain Monte Carlo methods. With the resulting parameter estimates, we can not only

[10]This assumption is not necessary for the model and can be replaced with a different distribution depending on context.

detect publication bias (through β), but we can also correct for the bias and estimate the underlying (true) admissibility rate. Technical details of the derivation can be found in Section 1.5.1.

1.2.4 Dependence Model

One of the key limitations of the independence model is that it assumes that a case's appearance on one list is conditionally independent of its appearance on another list. Violations of this assumption can result in underestimation. (Lum et al., 2013; Stanghellini and van der Heijden, 2004). For example, if one legal database copied its entries from another, then the resulting databases would exhibit more overlap than under true independence, biasing the model toward estimating fewer unobserved cases. A standard response to this problem is to include a group of interaction terms in the regression model, (Lum et al., 2013; Fienberg et al., 1999; Darroch et al., 1993), but such interaction terms greatly increase the complexity of the model and the difficulty of calculating the corrected admissibility rate.

Legal databases are arguably unlikely to suffer general interdependency. Legal databases cannot simply copy each other – the legal research market is lucrative, and providers rigorously police licensing agreements and intellectual property rights. This situation is in contrast to other contexts in which sharing might be actively promoted. Rather, most of the dependence among legal databases is likely to arise from joint dependence on one specific list—the "official" publication list. Historically, courts chose whether an opinion carried sufficient precedential importance to be printed in an official case reporter. Opinions not so selected were classified as "unpublished" and typically disappeared from public view since there was no easy access for future litigants. The advent of electronic databases has changed this dynamic, since electronic services include both officially "published" opinions as well as "unpublished" ones. Nonetheless, the published-unpublished distinction still carries legal heft. Officially published opinions generally carry greater precedential weight and are viewed as more important by legal actors.

The dependence seen in legal databases is thus of a specific, limited kind. Inclusion in an official reporter occurs independently, because the judicial decision to publish occurs ex ante. We will label publication of case k in an official reporter as Y_{kP} to match the previous notation of Y_{kl} for observing case k in list l. For notational simplicity, we will denote its complement as Z_{kP}, where $Z_{kP} = 1 - Y_{kP}$. Observation of case k in any of the remaining databases ($O_l, l = 1, \ldots, L$) is dependent on publication in the official reporter (O_P), but conditionally independent otherwise.

With these assumptions in place, we can model the probability that case k will be observed in list l with a slight modification to the regression model in (1.1):

$$\text{logit}(\phi_{kl}) = \beta A_k + \theta_k + \gamma_l + \rho_l Z_{kP} \tag{1.2}$$

where $l = 1, 2, \ldots, L, P$. The dependence of list l on the official publication list P is captured by the new parameter ρ_l, and to avoid circularity issues, ρ_P is set to zero. We assume that the ρ_l arise from a common normal distribution, but all other elements and their priors are as before in Equation (1.1).

Once again, β is the parameter of interest, because it captures the effect of a case's outcome (admissibility) on its observation by a list. $\hat{\beta}$ thus enables us to detect whether case publication bias is a problem.

Using Bayes Rule with this regression model, we can also estimate a corrected admissibility rate. A full derivation is found in Section 1.5.2, but the essential result is as follows:

Let

$$R_{l|A} = \text{logit}^{-1}(\hat{\beta} + \hat{\gamma}_l + \hat{\rho}_l),$$

and

$$R_{l|\bar{A}} = \text{logit}^{-1}(\hat{\gamma}_l + \hat{\rho}_l).$$

Then, if p_{cor} is the corrected admissibility rate and p_{obs} is the observed admissibility rate,

$$\frac{p_{cor}}{1 - p_{cor}} = \frac{\widehat{P(O^*|\bar{A})}}{\widehat{P(O^*|A)}} \frac{p_{obs}}{1 - p_{obs}},$$

where $\widehat{P(O^*|A)}$ is the real root $x \in (0,1)$ of

$$(1-x)(1 - R_{P|A}x)^{L-1} - \prod_{l=1}^{L}[1 - (R_{P|A} + R_{l|A} - R_{P|A}R_{l|A})x] = 0,$$

and $\widehat{P(O^*|\bar{A})}$ is the real root $x \in (0,1)$ of

$$(1-x)(1 - R_{P|\bar{A}}x)^{L-1} - \prod_{l=1}^{L}[1 - (R_{P|\bar{A}} + R_{l|\bar{A}} - R_{P|\bar{A}}R_{l|\bar{A}})x] = 0.$$

One final complication to the model involves case "types." As described in Section 1.1, we can observe trial court decisions either directly through the trial court's own opinion, or indirectly through subsequent appellate decisions. As one might expect, some court levels are more likely to be observed in legal databases than others. Supreme Court opinions are closely followed; trial court opinions are not; appellate court opinions are somewhere in between. We can revise the regression model further to account for the type of opinion (i.e., trial, appellate, etc.):

$$Y_{klt} \sim \text{Bernoulli}(\phi_{klt})$$

$$\text{logit}(\phi_{klt}) = \beta A_k + \theta_k + \gamma_l + \tau_t + \rho_l Z_{kP} \tag{1.3}$$

where τ_t captures the effect that the court's level t has on the probability that its opinion in case k is observed in list l. Roughly in line with legal experience, the model assumes that observing a case at one level of the court hierarchy is independent of observing it at another level. The parameter β in the model serves the same purpose of bias detection as in the earlier models. Calculating the corrected admissibility rate requires that we account for τ_t, the details of which are shown in Section 1.5.3. Otherwise, the derivation and result are effectively the same as for the previously discussed dependence model.

1.3 Applications

1.3.1 Simulated Dataset

To validate the full dependence model (with case types) discussed in Section 1.2.4, I constructed a series of simulated case publication datasets. Each simulation began with a population of 250 cases. In each case, the proffered evidence had a 33% chance of admission, which was the "true" admissibility rate. Six different publication databases were available to observe the cases at either the trial or appellate level. One list was designated the official publication list (P). Two of the remaining lists (L and W) were a combination of independent observations and copies from the official publication list,[11] and the final three lists (A, B, and C) consisted simply of independent observations. Aside from the defined dependence between L, W and P, all of the lists had conditionally independent draws. Table 1.2 reports the probabilities at which a given list l observed an admissibility or exclusion ruling at either the trial or appellate level.

	Probability of Observation			
List	Trial Opinion Admission	Trial Opinion Exclusion	App Opinion Admission	App Opinion Exclusion
P	$2p_{tr,ad}$	$2p_{tr,ex}$	$2p_{ap,ad}$	$2p_{ap,ex}$
A	$p_{tr,ad}$	$p_{tr,ex}$	$p_{ap,ad}$	$p_{ap,ex}$
B	$2p_{tr,ad}$	$2p_{tr,ex}$	$2p_{ap,ad}$	$2p_{ap,ex}$
C	$4p_{tr,ad}$	$4p_{tr,ex}$	$4p_{ap,ad}$	$4p_{ap,ex}$
L (L=L'+P)	$2p_{tr,ad}$	$2p_{tr,ex}$	$2p_{ap,ad}$	$2p_{ap,ex}$
W (W=W'+P)	$4p_{tr,ad}$	$4p_{tr,ex}$	$4p_{ap,ad}$	$4p_{ap,ex}$

Table 1.2: Simulated Lists and Their Observation Probabilities

[11] Lists L and W simulate the two major legal databases, Lexis and Westlaw, which comprehensively include the officially published cases.

The baseline probabilities of observing cases (i.e., $p_{tr,ad}$, $p_{tr,ex}$, $p_{ap,ad}$, and $p_{ap,ex}$), represent respectively the probability for trial cases with admission, trial cases with exclusion, appellate cases with admission, and appellate cases with exclusion. These baselines were varied over several runs to simulate no publication bias, moderate publication bias, strong publication bias, as well as publication bias only at the trial level or only at the appellate level. Keeping with actual practice, the probability of observation was always significantly higher at the appellate level than at the trial level. As seen in Table 1.3, despite the true admissibility rate being 33%, the observed admissibility rate departs increasingly downward from the true rate as one increases the publication bias. In addition, most of the observed rate is driven by appellate opinion observations, since relatively few trial court opinions are ever directly observed at all.

Simulated Set	$p_{tr,ad}$	$p_{tr,ex}$	$p_{ap,ad}$	$p_{ap,ex}$	Observed Admis Rate
No Bias	0.0025	0.0025	0.05	0.05	0.319
Moderate Bias	0.002	0.004	0.04	0.08	0.244
Strong Bias	0.001	0.005	0.02	0.10	0.137
Appellate Bias Only	0.003	0.003	0.04	0.08	0.247
Trial Bias Only	0.001	0.005	0.06	0.06	0.313

Table 1.3: Simulated Datasets Used

Using these simulated datasets, we estimated the model in Equation 1.3 via Markov Chain Monte Carlo (MCMC) methods to arrive at estimates for β and the corrected admissibility rate. The results are shown in Table 1.4. For point estimates, we used the mean of the posterior distribution for both β and the estimated corrected admissibility rate (p_{cor}). (p_{cor} was calculated for each draw of the MCMC sampler.)

Simulated Set	β	(95% CI)	Observed Admis	Corrected Admis
No Bias	0.0	(-0.4, 0.3)	0.319	0.334
Moderate Bias	-0.4	(-0.7, -0.1)	0.244	0.311
Strong Bias	-1.0	(-1.5, -0.6)	0.137	0.346
Appellate Bias Only	-0.4	(-0.7, 0.0)	0.247	0.306
Trial Bias Only	0.0	(-0.3, 0.3)	0.314	0.315

Table 1.4: Simulation Results (True Admissibility Rate = 0.333)

1.3.2 False Confession Expert Testimony

False confession expert testimony (FCET) is an important context for applying the bias-correction model. Confessions are, of course, one of the most devastating potential pieces of evidence against a criminal defendant. In recent years, however, wrongful conviction exonerations have shown surprisingly that defendants will at times confess to crimes that they did not commit. Social scientists have studied risk factors leading to false confessions, including defendant disposition, environment, and interrogator behavior, and some defendants have sought to introduce this type of testimony to lend credibility to their claim that they confessed falsely. (Kassin and Kiechel, 1996; Chojnacki et al., 2008; Ofshe and Leo, 1997).

As previously noted, FCET is a context in which one might be especially concerned about publication bias and its resulting distortions. FCET is relatively new, controversial, and involves psychological expertise, (Kaye et al., 2016, §§2.7.4 and 8.9.6), making judges likely to look to other sources, such as the case law, for guidance on its evidentiary reliability. The ability to conduct unbiased surveys of the case law is therefore critical. At the same time, because FCET involves confessions and takes place in the criminal context, it is especially at risk of the biasing mechanisms discussed in Section 1.1. At the trial level, excluding FCET severely hampers the defense, placing added pressure on trial judges to offer a written explanation when excluding the evidence. In addition, to the extent FCET is effective defense evidence, its admission will result in more acquittals, which cannot be appealed. Appellate opinions discussing FCET will therefore more likely arise out of trial court exclusions.

To study publication bias in the FCET context, cases dealing with FCET were gathered from readily available legal search databases: Westlaw, Lexis, Google Legal, BNA, FastCase, and Daubert Tracker.[12] Some of these databases are likely well-known to readers, others less so; some are fee-based subscription services, while others are freely available to the public. From these lists, one can easily extract the "official publication list," meaning cases appearing in traditional case reporters. Predictably, both the Westlaw and Lexis lists included all officially published cases. However, they also included scattered "unpublished" opinions that were not coincident with each other.

In total, the false confession dataset consisted of 136 case opinions, of which 13 were trial opinions, 95 were decisions from a direct appeal, and 28 were from further appellate processes (subsequent appeals, collateral review, etc.). These 136 case opinions derived from 110 underlying cases. The observed rate at which trial

[12] My research assistant, Maria Ramrath, a 2015 J.D./M.B.A. graduate of Vanderbilt University, primarily handled this "case survey," replicating what a junior associate might do in an actual litigation context. Cases were found using conventional methods in legal research: search terms (e.g., "'false confession' /s 'expert testimony'"), cross-checks between databases, and pursuing citations found in the search results and citation history. Cases were then read to determine if they were "relevant," meaning that they either discussed the admissibility of false confession expert testimony directly, or discussed an earlier trial court's opinion on admissibility.

courts admitted FCET was 0.16.

Confirming the suggestion in Section 1.1 that appellate courts tend to affirm trial evidentiary decisions, appellate deference in the FCET dataset was very high. Among the 114 appellate opinions that reached the issue of admissibility, 89% affirmed the trial court's determination.[13] This affirmance rate was high irrespective of whether the underlying trial court ruling was admission (82.3%) or exclusion (90.7%).

We estimated the publication bias correction model in Section 1.2.4 for the FCET dataset. MCMC estimates were made using the Stan statistical computing platform, and visual checks were made of the trace plots to ensure proper mixing. The results are found in Table 1.5. The results suggest that the published case law on FCET may exhibit a bias against admitted cases. The coefficient β, which is the indicator of bias, is indeed negative, although less dramatic than in the simulated datasets. In addition, the 95% credibility interval[14] includes zero. Given the small sample size and the noisier real-world data, the higher degree of uncertainty is unsurprising. Rather revealing, however, is the model's estimate for the actual (corrected) admissibility rate, which is 0.281, rather than the observed rate of 0.163. This result suggests that the case law could potentially be misleading legal actors into thinking that FCET is far less accepted in trial court rulings than in reality.

[13]This rate is for all appellate opinions. The rate for affirmance on the first direct appeal was a similar 90%.

[14]Because estimation of the parameters was through Bayesian techniques, the conventional measure of uncertainty is through a so-called "credibility" interval, as opposed to the confidence interval in classical statistics.

	β (95% CI)	Observed Admis	Corrected Admis (95% CI)
False Confessions Set	-0.211 (-0.702, 0.165)	0.163	0.281 (0.100, 0.794)

Table 1.5: Results from False Confession Expert Testimony Dataset

1.4 Discussion

1.4.1 Model Limitations and Assumptions

As with all statistical models, the case publication bias correction model presented here relies on a variety of assumptions to do its work. The discussion below outlines some of those assumptions and their accompanying limitations.

1.4.1.1 Sampling Independence and Coverage

One of the key requirements for the MSE technique is that the various lists be conditionally independent and that they have access to the entire underlying population. For instance, in the fish example, if certain fish dislike the bait used and can never be caught, then the procedure will systematically underestimate the population. (The proposed model addressed a related concern by accounting for official publication status, since cases that are officially published are more likely to be "caught" than those that are not.)

At first, oral admissibility decisions may seem to create a problem along these lines. Legal research databases do not typically include trial transcripts, so oral decisions would seem to be a permanently hidden population. Fortunately for our purposes, however, subsequent appeals generate opinions, and these opinions can shed light on the otherwise hidden world. Further, at least in the case of FCET testimony, the bias created by these hidden oral admissibility decisions arguably cut in a "conservative" direction. As Section 1.1 suggests, judges may be more likely to resort to oral decisions when admitting, rather than excluding FCET evidence, at least as an initial matter.[15] A failure of the lists to capture oral decisions would thus result in the model underestimating the population of admissions and hence "undercorrect" the admissibility rate. So perhaps the admissibility rate estimated by the model is in actuality a floor, but in any event is still better than the uncorrected admissibility rate.

1.4.1.2 List Similarity

The proposed model requires that all of the publication lists suffer from the same biasing mechanism(s). This requirement comes from the model's roots in multiple systems estimation, which uses overlaps and gaps in the publication lists to infer the existence of unobserved cases. (The proposed model links two conventional MSE models together — an MSE model for excluded cases and an MSE model for admitted cases.) This limitation means that one cannot mix legal research database lists with, for example, partisan lists compiled by attorneys or experts.

To illustrate, suppose one added a partisan list that included only admissions. This list would then sharply increase the estimated number of admission cases, since many of the partisan list's cases would not be found in the other lists, implying a

[15] As discussed below, as the body of case law matures however, the direction of the bias may shift, because novelty may motivate the publication of written opinions.

significant hidden population. At the same time, however, the estimated number of excluded cases would change less dramatically, since the supplemental dataset contributes little new information on exclusions. Thus, by manipulating the size of this added partisan list, one could arbitrary manipulate the level of publication bias observed.[16] Fortunately, there is no reason to believe that the biasing mechanism among the various legal research database varies, and certainly no reason to believe they are purposely biased in partisan ways. The assumption required by MSE is thus reasonable.

1.4.1.3 Focus on Trial Court Admissibility

The proposed model focuses on trial court admissibility determinations, which can be observed either through the original trial court's opinion or indirectly through subsequent appellate opinions. This focus on trial decisions may seem initially odd, since a subsequent appellate ruling could reverse and supersede the trial court. Appellate opinions also typically have broader precedential weight within their jurisdiction. However, given the abuse-of-discretion standard that governs evidentiary rulings, appellate opinions have a more limited role in evidence than in other areas of law. If an appellate court affirms, the appellate decision has only modest precedential effect, since the appellate court has only stated deferentially that a trial court *may in its discretion* admit/exclude in that context.[17] Appellate reversals are more weighty

[16] Additional simulation studies done on the bias correction model confirm these arguments, although for brevity, we will not discuss them further here.

[17] While an appellate court could decide to adopt a per se rule, these are typically rare events. One traditional area in which courts have imposed per se inadmissibility is polygraphs, (Ligons, 2000), but the modern trend is not toward such rules.

precedent, but as a practical matter they are quite rare. As reported above, for the FCET dataset, trial court rulings had a $\sim 90\%$ affirmance rate. Furthermore, our objective in this study is not to determine binding law. Rather, its more limited goal is to discover whether the standard processes of publication and appeal contribute to a skewing of the rulings that are observed. We are concerned with the psychological effect the skewed pool might have on legal actors.

1.4.1.4 Biasing Mechanism

The current model collapses the different biasing mechanisms that might be at play into a single parameter, β. So for example in the FCET context, it does not separate the bias created by a trial judge's felt-obligation to explain pro-prosecution rulings from the bias caused by double jeopardy rules. Conceptually, one can construct a model that treats each hierarchical level (trial, appellate, etc.) separately. For example:

$$Y_{kl}^T \sim \text{Bernoulli}(\phi_{kl}^T)$$

$$\text{logit}(\phi_{kl}^T) = \beta^T A_k + \theta_k + \gamma_l$$

$$Y_{kl}^A \sim \text{Bernoulli}(\phi_{kl}^A)$$

$$\text{logit}(\phi_{kl}^A) = \beta^A A_k + \theta_k + \gamma_l$$

where the T and A superscripts denote data or parameters specifically dealing with observations at the trial or appellate level. This expanded model can grow to accommodate additional levels of appellate process, and differs from Equation 1.3, where we had a single bias parameter (β) and then other parameters controlling for type (τ). Given the small numbers of trial court opinions in our FCET dataset, we did not pursue this model further.

1.4.1.5 Case Temporal Independence

The proposed model accounts for some list interdependence (more specifically, their dependence on official publication), but it assumes that the cases are temporally independent, meaning that their probability of observation does not depend on when they occur. This assumption is arguably violated in practice.[18] For example, early cases or cases of first impression in a jurisdiction are more likely to generate legal community interest, making inclusion in a database more likely. Worse yet, the biasing effect of admissibility may change over time depending on state of the corpus of published cases. A judge may be more likely to write and publish a decision when it goes "against the grain." Presumably, covariates could account for these nuances (albeit crudely), but modeling is practically difficult because of the small sample sizes, large number of jurisdictions, the hierarchical nature of the judiciary, and the complex rules about precedent. The model's use of case-specific random effect terms alleviates some of these concerns — while it does not model temporality directly, the model does account for case-by-case variation in observability.

Accounting for temporal independence could be a useful area for future research. A big concern with legal publication is its potential to create cascade effects. If later legal decisions are more likely to conform with earlier ones, and the observation of

[18]My thanks to Paul Edelman for raising this point in detail.

earlier decisions is biased, then the effect can snowball. For example, suppose at time 1, decisions are half admissions and half exclusions. Because of publication bias, however, at time 2, an observer sees primarily exclusions and is therefore more likely to exclude. Then, at time 3, due to both publication bias and the growing cascade effect, an observer will start to see more and more "consensus" toward exclusion, even though the consensus is an artifact of publication bias.

1.4.2 Policy Implications

The policy implications for the ideas explored in this paper are wide-ranging. To the extent that academics, practitioners, and courts conduct informal case reviews, they must proceed cautiously, cognizant of the possibility that observed cases may not be representative of the underlying population. Perhaps more importantly, for more sophisticated actors aware of case publication bias but still desiring to use published databases, this Chapter has offered an approach and a useful model. It provides a method for detecting and then subsequently correcting publication bias, provided that multiple legal research databases are available.

More specifically, the application of the model to the false confession expert testimony context yields concerning results that deserve further investigation. The model suggests that the observable case law gives a distorted view of the frequency with which courts admit FCET. An attorney or judge naively looking at available legal databases would conclude that FCET has been poorly received (16%), whereas the model suggests greater ambivalence among trial courts (28%). Indeed, if FCET admissibility rulings have experienced a negative cascade effect (as described above), the underlying "true" admissibility rate might be even higher.

Finally, while the proposed bias-correction model may be an exciting and powerful tool for addressing case publication bias, legal actors should ideally only look at it as a temporary solution. Complicated statistical models are often necessary only because the available data is deeply flawed, and their results, while helpful, come at the expense of many assumptions and uncertainty. Far better would be to have datasets without such flaws, obviating the need for the models entirely. One hope is that greater awareness of case publication bias will lead to more comprehensive availability of legal decisions. For example, trial judges might try to write and release opinions regardless of outcome. State court systems could make trial transcripts more widely available, and legal research databases could try to obtain case materials in more systematic ways. For example, Lex Machina, the legal analytics company spawned from Stanford's law school and computer science department, has attempted to independently collect its own exhaustive dataset on intellectual property cases. If these kinds of databases were more pervasively used, the bias-correction model might be unnecessary, and that would be a good thing.[19]

[19] A more "legal" solution would be to encourage greater uniformity of evidentiary determinations through appellate practice reforms. For example, removing the abuse-of-discretion standard would move the battleground of evidence law to the appellate level, where opinions are more readily accessible. More radically, one can envision allowing the government to appeal acquittals on an advisory basis, although courts have largely disfavored this option. (See Stith, 1990, 7 & n.14).

1.5 Technical Derivations

1.5.1 Independence Model

Table 1.6 provides a reference key for all the notation used in this Section 1.5.1.

Variable	Definition
k	Index for cases (1, ..., K)
l	Index for lists (1, ..., L)
Y_{kl}	Observation of case k in list l
A_k	Expert admitted in case k
ϕ_{kl}	Probability that case k observed in list l
β	Effect of A_k on observation
θ_k	Random-effect term for case k
γ_l	Random-effect term for list l
O_l	Observation in list l
O^*	Observation in any list
w_k	Normalization term

Table 1.6: Reference Key for Independence Model

Recall that for the Independence Model, we model the observation of case k in list l through the following regression model:

$$Y_{kl} \sim \text{Bernoulli}(\phi_{kl})$$

$$\text{logit}(\phi_{kl}) = \beta A_k + \theta_k + \gamma_l \tag{1.1}$$

$$\theta_k \sim \mathcal{N}(0, \sigma_\theta^2)$$

To find the corrected admissibility rate, we set up the problem using Bayes Rule, where O_l represents observation in list l, O^* represents observation in any available list, and A ($\bar{A}$) is a ruling in favor of admissibility (exclusion).

$$\frac{P(A|O^*)}{P(\bar{A}|O^*)} = \frac{P(O^*|A)}{P(O^*|\bar{A})}\frac{P(A)}{P(\bar{A})}$$

Immediately we can identify several terms. The prior odds ($P(A)/P(\bar{A})$) are the unknown quantity of interest, because $P(A)$ is the corrected admissibility rate unencumbered by publication bias. The posterior odds ($P(A|O^*)/P(\bar{A}|O^*)$) are easily estimated by using the observed (uncorrected) admissibility rate. Legal observers only see cases that are reported in some list and calculate the admissibility rate from that potentially skewed perspective. The likelihood ratio ($P(O^*|A)/P(O^*|\bar{A})$) can ultimately be inferred from the estimated regression coefficients in the model, but doing so requires further elaboration.

Without loss of generality, we work with the numerator, $P(O^*|A)$. Owing to the fact that $O^* = \bigcup_l O_l$, and the model assumption that the various lists are independent:

$$P(O^*|A) = 1 - \prod_l (1 - P(O_l|A)) \tag{1.4}$$

Next, by the definition of conditional probability,

$$P(O^*|AO_l)P(O_l|A) = P(O_l|AO^*)P(O^*|A),$$

where $P(AO_l)$ is shorthand for $P(A \cap O_l)$. But since $O^* = \bigcup_l O_l$, $P(O^*|AO_l) = 1$, we have

$$P(O_l|A) = P(O_l|AO^*)P(O^*|A). \tag{1.5}$$

Substituting (1.5) into (1.4),

$$P(O^*|A) = 1 - \prod_l (1 - P(O_l|AO^*)P(O^*|A)).$$

This last relationship is key because the regression coefficients from the model are estimates of $P(O_l|AO^*)$, not $P(O_l|A)$. The model estimates the probability of observing a case in list l given that the case is observed in some list (O^*). We now have an estimator for $P(O^*|A)$, which is implicitly expressed in the equation:

$$\widehat{P(O^*|A)} = 1 - \prod_l (1 - \text{logit}^{-1}(\beta + \theta_k + \gamma_l)\widehat{P(O^*|A)}))$$

Or more conveniently, $P(O^*|A)$ is a root x of the polynomial:

$$1 - x - \prod_l (1 - \text{logit}^{-1}(\beta + \theta_k + \gamma_l)x)) = 0$$

This polynomial has a trivial root at zero, as well as a real root on (0,1) whenever $\sum_l P(\widehat{O_l|AO}^*) \geq 1$, a condition that will be satisfied whenever we have reasonable estimates for $P(\widehat{O_l|AO}^*)$. Since the individual lists are independent, and the case must appear on some list, then the sum of the list probabilities should be at least one. Higher order polynomials will have additional real or complex roots, but those appear to all have real part greater than 1, so we will ignore them.

One last complication is that the regression contains a nuisance parameter θ_k, the case random effect term. Since θ_k parameters are normally distributed around zero, we can arguably treat it as zero for purposes of estimation, which worked well in simulation runs. Alternatively, one can integrate it out formally:

$$P(O_l|AO^*) = \sum_k P(O_l|kAO^*)P(k|AO^*) \tag{1.6}$$

The first term in (1.6) can be estimated with the regression model:

$$P(\widehat{O_l|kAO^*}) = \text{logit}^{-1}(\beta + \theta_k + \gamma_l), \quad l = 1, \dots, L$$

The second term in (1.6), $P(k|AO^*)$, can be estimated directly from the dataset. This probability is uniform over all cases with admissible outcomes, and zero for all cases with exclusion outcomes. Consolidating, we have

$$P(\widehat{O_l|AO^*}) = \sum_k w_k \, \text{logit}^{-1}(\beta + \theta_k + \gamma_l), \quad l = 1, \dots, L$$

where

$$w_k = \begin{cases} \frac{1}{\sum_m A_m}, & \text{if } A_k = 1 \\ 0, & \text{otherwise} \end{cases}.$$

So the revised estimator for $P(O^*|A)$ becomes implicitly expressed in the equation:

$$P(\widehat{O^*|A}) = 1 - \prod_l (1 - (\sum_k w_k \, \text{logit}^{-1}(\beta + \theta_k + \gamma_l) P(\widehat{O^*|A})))$$

1.5.2 Dependence Model

Table 1.7 provides a reference key for all of the notation used in Section 1.5.2.

Variable	Definition
k	Index for cases (1, ..., K)
l	Index for lists (1, ..., L, P)
Y_{kl}	Observation of case k in list l
A_k	Expert admitted in case k
Z_{kP}	Whether case k is (officially) unpublished
ϕ_{kl}	Probability that case k observed in list l
β	Effect of A_k on observation
θ_k	Random-effect term for case k
γ_l	Random-effect term for list l
ρ_l	Effect of being (officially) unpublished on observation in list l
O_l	Observation in list l
O_P	Observation in official publication list (P)
O^*	Observation in any list
w_k	Normalization term
$R_{l\|A}$	Probability of being observed in list l given expert is admitted
x	Root of Equation 1.10

Table 1.7: Reference Key for Dependence Model

As discussed in Section 1.2.4, we assume for this model that publication in an official reporter is an independent determination. Observation in any other database is influenced by (i.e., dependent on) official publication, but conditionally independent otherwise. Using this structure, we can obtain an estimate for the corrected admissibility rate.

We again begin with Bayes Rule:

$$\frac{P(A|O^*)}{P(\bar{A}|O^*)} = \frac{P(O^*|A)}{P(O^*|\bar{A})}\frac{P(A)}{P(\bar{A})} \tag{1.7}$$

As with the independence model, the prior odds are unknown, but are the quantity of interest. The posterior odds are easily estimated by using the observed (uncorrected) admissibility rate. The key to estimating the corrected admissibility rate is the likelihood ratio $(P(O^*|A)/P(O^*|\bar{A}))$. Without loss of generality, we again focus on the numerator $P(O^*|A)$.

Owing to the fact that $O^* = \bigcup_l O_l$ and that the lists are conditionally independent given O_P, we can use deMorgan's Law and the product rule:

$$\begin{aligned} P(\bar{O^*}|A) &= P(\bar{O_1} \cap \bar{O_2} \cap \dots \bar{O_L} \cap \bar{O_P}|A) \\ &= P(\bar{O_1} \cap \bar{O_2} \cap \dots \bar{O_L}|\bar{O_P}, A)P(\bar{O_P}|A) \\ &= P(\bar{O_1}|\bar{O_P}, A)P(\bar{O_2}|\bar{O_P}, A) \dots P(\bar{O_L}|\bar{O_P}, A)P(\bar{O_P}|A). \end{aligned}$$

To get an estimate for $P(\bar{O_l}|\bar{O_P}, A), l = 1, 2, \dots L$, we note that

$$P(O_l, O^*|\bar{O_P}, A) = P(O_l|O^*, \bar{O_P}, A)P(O^*|\bar{O_P}, A)$$

but because $O_l \subseteq O^*$,

$$P(O_l|\bar{O_P}, A) = P(O_l|O^*, \bar{O_P}, A)P(O^*|\bar{O_P}, A) \tag{1.8}$$

We can estimate $P(O_l|O^*, \bar{O_P}, A)$ by using the modified regression model in Equation (1.2):

$$\text{logit}(\phi_{kl}) = \beta A_k + \theta_k + \gamma_l + \rho_l Z_{kP} \tag{1.2}$$

where $l = 1, 2, \dots, L, P$. Recall that here, $Z_{kP} = 1 - Y_{kP}$, that is, Z_{kP} is the complement of Y_{kP}, which is whether case k is observed on the official publication list P. The dependence of list l on the official publication list P is captured by the new parameter ρ_l, and to avoid circularity issues, ρ_P is set to zero. All other elements are as before in Equation (1.1).

Estimates for $P(O_l|O^*, \bar{O_P}, A)$ flow directly from the new model. To simplify the notation, we will label these estimates $R_{l|A}$:

$$R_{l|A} = P(O_l|\widehat{O^*, \bar{O}}_P, A) = \text{logit}^{-1}(\beta + \gamma_l + \rho_l)$$

where again the nuisance parameter, θ_k, can be treated as zero or integrated out formally. Note that because the probability is conditioned on A, $A_k = 1$, and because it is conditioned on $\bar{O_P}$, $Z_{kP} = 1$.

Finally, we analyze the remaining term in (1.8), which is $P(O^*|\bar{O_P}, A)$.

$$\begin{aligned}
P(O^*|\bar{O_P}, A) &= 1 - P(\bar{O^*}|\bar{O_P}, A) \\
&= 1 - \frac{P(\bar{O^*}\bar{O_P}|A)}{P(\bar{O_P}|A)} \\
&= 1 - \frac{P(\bar{O^*}|A)}{P(\bar{O_P}|A)}, && \text{since } \bar{O^*} \subseteq \bar{O_P} \\
&= \frac{P(\bar{O_P}|A) - P(\bar{O^*}|A)}{P(\bar{O_P}|A)} \\
&= \frac{P(O^*|A) - P(O_P|A)}{1 - P(O_P|A)}. && (1.9)
\end{aligned}$$

Since $O_P \subseteq O^*$,

$$\begin{aligned}
P(O_P|A) &= P(O_P O^*|A) \\
&= P(O_P|O^*, A)P(O^*|A).
\end{aligned}$$

$P(O_P|O^*, A)$ can be similarly estimated using the regression model in (1.2) with

$l = P$. For notational simplicity, we will label that $R_{P|A}$.

$$R_{P|A} = P(\widehat{O_P|O^*}, A) = \text{logit}^{-1}(\beta + \gamma_P),$$

where ρ has been dropped because $\rho_P = 0$. Substituting these results back into (1.9):

$$\begin{aligned} P(\widehat{O^*|\bar{O}_P}, A) &= \frac{P(O^*|A) - R_{P|A}P(O^*|A)}{1 - R_{P|A}P(O^*|A)} \\ &= P(O^*|A)\frac{1 - R_{P|A}}{1 - R_{P|A}P(O^*|A)}. \end{aligned}$$

Substituting back into (1.8):

$$\begin{aligned} P(\widehat{O_l|\bar{O}_P}, A) &= P(O_l|O^*, \bar{O}_P, A)P(O^*|\bar{O}_P, A) \\ &= R_{l|A}P(O^*|A)\frac{1 - R_{P|A}}{1 - R_{P|A}P(O^*|A)} \end{aligned}$$

We then substitute these results back into (1.7) to arrive at:

$$\begin{aligned} P(\bar{O}^*|A) &= P(\bar{O}_P|A)\prod_{l=1}^{L} P(\bar{O}_l|\bar{O}_P, A) \\ P(O^*|A) &= 1 - P(\bar{O}_P|A)\prod_{l=1}^{L}(1 - P(O_l|\bar{O}_P, A)) \\ &= 1 - \big(1 - R_{P|A}P(O^*|A)\big)\prod_{l=1}^{L}\left[1 - R_{l|A}P(O^*|A)\frac{1 - R_{P|A}}{1 - R_{P|A}P(O^*|A)}\right] \\ &= 1 - \big(1 - R_{P|A}P(O^*|A)\big)\prod_{l=1}^{L}\left[1 - R_{l|A}P(O^*|A)\frac{1 - R_{P|A}}{1 - R_{P|A}P(O^*|A)}\right] \end{aligned}$$

Our estimator $\widehat{P(O^*|A)}$ is thus the real root x on the interval (0,1) of the polynomial:

$$1 - x - \left(1 - R_{P|A}x\right) \prod_{l=1}^{L} \left[1 - R_{l|A}x \frac{1 - R_{P|A}}{1 - R_{P|A}x}\right] = 0$$

or after simplification,

$$(1 - x)(1 - R_{P|A}x)^{L-1} - \prod_{l=1}^{L} [1 - (R_{P|A} + R_{l|A} - R_{P|A}R_{l|A})x] = 0. \tag{1.10}$$

1.5.3 Dependence Model with Types

Section 1.5.3 uses the notation in Table 1.7 as well as Table 1.8 brlow.

Variable	Definition
t	Index for case types (trial or appellate)
τ_t	Random-effect term for case type t
T_t	Type t court generated an opinion in the case
O_{lt}	Observation of type t court opinion in case

Table 1.8: Additional Reference Key for Dependence Model with Types

Recall that in Equation 1.3, we revised the regression model further to account for the type of opinion (i.e., trial, appellate, etc.):

$$Y_{klt} \sim \text{Bernoulli}(\phi_{klt})$$

$$\text{logit}(\phi_{klt}) = \beta A_k + \theta_k + \gamma_l + \tau_t + \rho_l Z_{kP} \tag{1.3}$$

where τ_t captures the effect that the court's level t has on the probability that its opinion in case k is observed in list l.

To calculate the corrected admissibility rate, we need to account for τ_t. In the following analysis, all probabilities are dependent on admissibility (A) and observation in the dataset (O^*), but we will drop the dependence notation for simplicity. Let $P(O_{lt})$ be the probability that a type t court's opinion in case k is observed on list l, and $P(O_l)$ be the probability the case k is observed on list l at any court level. Then,

$$
\begin{aligned}
P(O_l) &= P(O_{l1}T_1 \cup O_{l2}T_2 \cup \ldots \cup O_{lT}T_T) \\
&= \sum_t P(O_{lt}T_t) - \sum_t \sum_{s<t} P(O_{lt}O_{ls}T_tT_s) + \ldots \\
&= \sum_t P(O_{lt}|T_t)P(T_t) - \sum_t \sum_{s<t} P(O_{lt}O_{ls}|T_tT_s)P(T_tT_s) + \ldots \qquad (1.11)
\end{aligned}
$$

where T_t denotes the event that a type t court generated an opinion in the case.

We can readily estimate all of the values in (1.11). For the first-order summation, $P(O_{lt}|T_t)$ comes directly from the regression model, since

$$\widehat{P(O_{lt}|T_t)} = \text{logit}^{-1}(\beta + \gamma_l + \tau_t)$$

where again we have assumed either that $\theta_k \approx 0$ or we can integrate it out. We can estimate $P(T_t)$ directly from the dataset matrix by using the frequency at which observed admissible cases are from level t (as opposed to other court levels).

For the higher-order summations, we note that under the model assumptions, what happens to a case at the t level is independent of what happens to it at other levels, so

$$O_{lt} \perp\!\!\!\perp T_s, \quad t \neq s$$

$$O_{lt} \perp\!\!\!\perp O_{ls}, \quad t \neq s$$

and so we can simplify the joint probabilities using the product rule. For example, for the second-order summation:

$$\sum_{t} \sum_{s<t} P(O_{lt} O_{ls} | T_t T_s) P(T_t T_s) = \sum_{t} \sum_{s<t} P(O_{lt} | T_t) P(O_{ls} | T_s) P(T_t T_s).$$

$P(O_{lt}|T_t)$ and $P(O_{ls}|T_s)$ can once again be estimated using the regression model, whereas $P(T_t T_s)$ can be estimated from the dataset matrix by looking at frequencies.

Having thus found a way to estimate $P(O_l|O^*, A)$, the rest of the method for correcting the observed admissibility rate in the dependence-type model proceeds as in Section 1.5.2.

2 Fair Division of Attorneys' Fees[1]

At the end of every class action or consolidated multidistrict settlement, a court must decide how much to pay the group of lawyers who the court appointed to prosecute the case on behalf of the plaintiffs. Because aggregate litigation lawyers are appointed by courts and not selected by clients in a free market like other lawyers, there is no possibility of paying the lawyers using voluntarily (ex ante) contractual arrangements; the court must figure out how to do it ex post. As such, a great deal of legal scholarship has been published to help courts decide how much to pay these groups in total. (Fitzpatrick, 2010c,a; Eisenberg and Miller, 2010; Rubenstein, 2009). But there is almost no scholarship on the question that inevitably follows: how should the money be divided amongst the group?

Most of the time, this task falls to the lawyer who was appointed by the court to be the leader of the group of lawyers, but sometimes the court tries to do it itself — and if a dispute arises when the lead lawyer does it, then the court will have to do it itself.[2] Without guidance from scholars, lead lawyers and courts rely upon a familiar technique: the lodestar method. That is, almost all fee allocations start with how many hours each lawyer says he or she worked on the case multiplied by that lawyer's normal hourly rate. Lead lawyers and courts usually do not stop there — they usually attempt to adjust those hours with "multipliers" that pay "more important" hours

[1]This chapter is joint work with Paul Edelman and Brian Fitzpatrick.

[2]For examples in which courts did the allocation themselves, see Allapattah Services, Inc. v. Exxon Corp. (2006) and In re TFT-LCD (Flat Panel) Antitrust Litigation (2013). For instances in which the lead lawyer did the allocation with review on objection by the court for abuse of discretion, see Victor v. Argent Classic Convertible Arbitrage Fund L.P. (2010), In re Initial Public Offering Securities Litigation (2011), and In re Vitamins Antitrust Litigation (2005).

more handsomely than "less important" ones — but the lodestar is almost always the foundation of every allocation. For example, in the recent NFL concussion litigation, the lead class action lawyer proposed dividing over $100 million in fees among a dozen or so law firms using this adjusted lodestar method, with multipliers ranging from 0.75 times the lowest firm's lodestar to 3.885 times the highest firm's lodestar (the lead lawyer's firm).

The problem with the lodestar method is that no one likes it. The vices of the lodestar method are well known in law and economics: if lawyers are paid for the hours they work, they will work more hours than necessary or they will pad their time by saying they worked more hours than they did. (Fitzpatrick, 2010a, 2051-52). They will also be insensitive to how much they recover for their clients because, so long as they recover something, they get paid the same. (Fitzpatrick, 2010a, 2051-52). These vices may not be very acute when a sophisticated client is around to supervise the lawyers, but that is not the case with class action and consolidated multidistrict litigation: the lawyers there are hired by courts, not clients, and courts have too many things to do and are institutionally ill-suited to watch over lawyers like a sophisticated client would. Our courts are good at adjudicating adversarial disputes that come to them, not reaching out and finding problems to solve on their own.

For these very reasons, courts today almost never use the lodestar method to determine the *total* attorneys' fee awarded. Fitzpatrick (2010c), for example, found that the lodestar method was used in only 12% of class action settlements. This has not always been the case. Before the law and economics scholarship critical of the lodestar method was published, most courts used the lodestar method to award fees after successful class action settlements. The famous 1985 Third Circuit Task Force Report authored by Arthur Miller brought much of this scholarship to the judiciary and was instrumental in the change in favor of the so-called "percentage method." (Fitzpatrick, 2010a, 2051-2052). Today, under the percentage method, courts pay the lawyers (as a group) a percentage of what they recover — typically 25% in class actions (Fitzpatrick, 2010c, 836) and 4% in multidistrict litigation (where the lawyers also collect a percentage from their individual clients) (Rubenstein, 2009, 88). This arrangement eliminates any incentive to drag cases along to generate unnecessary hours, and it makes the lawyers very sensitive to how much they recover for their clients.

But the percentage method is hard to implement when it comes to the *division* of fees. How do you decide which lawyer should get what portion of the common pool? If you give every lawyer the same percentage, what incentive would any of them have to do any work? If you try to award percentages based on how much you think each lawyer contributed to the outcome of the case, how do you figure out how much each lawyer contributed? The lawyers themselves — including the lead lawyers — have biased perspectives on how much they contributed, and as we noted, courts typically do not have very good information on their own regarding which lawyer did what. Thus, almost everyone falls back on the lodestar method for dividing the pool despite its obvious vices. (Baker et al., 2016).

The only better proposal (at least of which we are aware) is Silver and Miller (2010) in the context of consolidated multidistrict litigation. Their solution is basically to manufacture a sophisticated client: they propose tasking the lawyer with the greatest number of clients in the litigation with hiring the lawyers who will do all the work on behalf of the plaintiffs and figuring out how much to pay them; the catch is that the lawyer with the greatest number of clients cannot hire his or her own firm to do this work. They argue that the lawyer with the greatest number clients will have the incentive to hire the best people at the lowest price because that lawyer: 1) benefits more than any other from good work (he or she has the most individual clients, all paying him or her 33–40% in contingency fees), and 2) pays more for that work (because he or she is likewise paying more of the 4% common-benefit fees).

It is an ingenious solution, but it only works in contexts where you can manufacture a sophisticated client like this. In many consolidated multidistrict litigations, no single law firm has that many clients, and in class action litigation very few people have lawyers at all. Moreover, allocating fees among lawyers in aggregate litigation is not the only place the law confronts the problem of allocating money among a group of claimants ex post when there is no objective measure of proper distribution, but very strong subjective (and conflicting) beliefs. The same task confronts law firms when the partners allocate profits at the end of each year, when business associations unwind, and when inventors and artists jointly create great works. In these contexts, too, there is no easy way to manufacture a sophisticated client to make the division ex post. Sometimes ex ante contracts govern these divisions, but often they do not. How do we do it ex post?

In this Chapter, we tackle this question of proper distribution. Specifically, given a group of firms involved in a litigation, we propose asking each firm to rate the relative contribution made by the other firms with which they worked, and then using this information to reconstruct what is effectively a "consensus" division. Obviously, this consensus division will deviate some from each individual rater's preferences. In addition, not all firms will be familiar with the work of all other firms, and two firms may attempt to collude to inflate each other's share. These are some of the issues that we attempt to address below.

2.1 Background

The problem of how a group of claimants can equitably allocate a good among themselves is a fundamental one in any society. One classic technique, the "I cut, you choose" rule dates back at least 2800 years to Hesiod's *Theogeny*. (Brams and Taylor, 1996, 10). It is then no surprise that the literature, both academic and popular, is voluminous. (Brams and Taylor, 1996; Robertson and Webb, 1998; Moulin, 1991). With very few exceptions, however, the studies have focused on the situation in which each claimant's right is taken as a given. That is, all the parties agree to the value of the percentage claim of the other parties. The fundamental problem is how to design division rules that ensure that every party feels as if they were allocated their agreed upon share. For example, in the cake cutting problem, both parties agree that each deserves half the cake; the problem is only how to cut it in "half."

Our context, fee-splitting in class actions or multidistrict litigation, differs from other fair division questions on exactly this point. Because of the difficulties outlined in the Introduction, there is precisely no agreement on how much of the pot each firm merits. Thus, any mechanism to allocate the fees among the firms must decide, explicitly or tacitly, how much each is owed. Surprisingly, until recently, there has been little work in fair division on how to assess these values.

The first paper to consider this problem is de Clippel et al. (2008).[3] Their division rule is based on individual reports by each firm of the relative shares that each of the other firms deserve. They then propose three requirements that a division rule should satisfy:

1. (Strategy-proofness) The share of any firm is determined exclusively by the reports of the other firms — its own report has no influence on its own share.
2. (Objectivity) No firm's share is dependent on what a particular firm reports about its own share.[4]
3. (Consensuality) If there is a division that is consistent with all of the reports of all of the firms then that should be the final allocation.[5]

The principal result of de Clippel et al. (2008) is that there is a unique division rule for three firms that satisfies these three requirements. In particular, if s_i represents the share allocated to firm i, and r_{ij}^k is Firm k's report of the ratio of what Firm i and Firm j should receive, then

$$s_1 = \frac{1}{1 + r_{31}^2 + r_{21}^3}, s_2 = \frac{1}{1 + r_{32}^1 + r_{12}^3}, s_3 = \frac{1}{1 + r_{23}^1 + r_{13}^2}.$$

[3] For a more expository presentation of the same results, see Tideman and Plassmann (2008).

[4] B's share is not dependent on any report that A makes about A's value relative to anyone else.

[5] This terminology is taken from Tideman and Plassmann (2008).

(de Clippel et al., 2008, Prop.1).

An example of the de Clippel rule is helpful. Suppose that there are three firms, A, B, and C. Firm A reports that B's work was worth twice C's. Firm B reports that A was worth 3 times C, and C reports that A was worth 3/2's B. Given this information, how should we divide up the pot? In this instance, there is a division that is consistent with all three of the reports: A gets 1/2, B gets 1/3, and C gets 1/6.[6] Thus, the property of consensuality would require exactly that allocation.

Suppose instead that Firm A reports that B's work was worth twice C's, Firm B reports that A was worth 3 times C, and C reports that A was worth the same as B. Now, there is no division that is consistent with these three reports.[7] In this case, the de Clippel rule produces the allocation of A=3/7, B=2/5, and C=1/6. (de Clippel et al., 2008, Prop.1)

The astute observer will note that the sum of the shares of the firms do not add to 1.[8] That is, the de Clippel rule is not efficient in the sense that it forces some of the good to go unallocated. This would seem to be problematic. One could of course normalize the allocations, but the normalized allocation would no longer be guaranteed to satisfy the axioms. As it turns out, however, if there are four or more firms, a generalization of the de Clippel rule will in fact allocate all of the good.[9]

[6] $1/2 = 3 \times 1/6 = 3/2 \times 1/3$ and $1/3 = 2 \times 1/6$.

[7] To see this note that by C's report $\text{Share}_A/\text{Share}_B = 1$, and by A's report the $\text{Share}_B/\text{Share}_C = 2$. But then it follows that $\text{Share}_A/\text{Share}_C = \text{Share}_A/\text{Share}_B \times \text{Share}_B/\text{Share}_C = 1 \times 2 = 2$ which is inconsistent with B's report.

[8] $3/7 + 2/5 + 1/6 \approx 0.995$

[9] Unfortunately, if there are more than four firms then the rule is no longer unique, i.e., there are many rules, all satisfying the three given conditions, and all of which will be efficient. (de Clippel et al., 2008, Thm.2).

We could, given the results of de Clippel, et al., advocate for the use of the de Clippel rule in the context of fee-splitting in class actions. However, as beautiful as their work is, we find that it has some significant drawbacks in the fee-splitting context. The first problem is that it is rather difficult to motivate and explain the division rule. While one can appeal to the axioms and the theorems, the intuition behind the construction is somewhat obscure. To be practical, one has to be able to persuade the firms that the division rule is an improvement over the status quo, and the difficulty in doing so is a major impediment to its practical usefulness. We would expect that firms examining the difference between their desired portions and the final assignment will want some explanation for why the outcomes are different from what they recommended, and the de Clippel rule does not provide such an explanation.

Second, the mathematical elegance of the de Clippel rule — that there is a unique division rule that satisfies the three axioms — often breaks down in practice. In the fee division context, incomplete data is common. Firms A and B may work with Firm C, but not with each other, a situation that is particularly frequent when the number of firms involved grows large. To use the de Clippel rule, one must have at least one report for every pair of firms, a condition that is more readily violated than may seem at first glance (for example, see Table 2.1 *infra*). (Tideman and Plassmann, 2008) If one wants to use the version of the de Clippel rule that avoids unallocated funds, one must have at least *two* reports for every pair of firms. Thus, for example, in a division problem with Firms A, B, C, and D, to guarantee all three axioms, at least two firms must rate AB, AC, AD, BC, BD, and CD; in other words, no missing data is allowed.[10]

Finally, the opacity of the de Clippel rule leads to one further disadvantage. If collusion is a concern, it would be useful to see how each firm's reports contribute to the outcome. If one firm's reports are vast outliers, or if a pair of firms seem to be too cozy, it would be useful to see it overtly in the distribution rule. Such transparency may forestall such behavior. And if firms collude anyway, perhaps it can be detected and the firms suitably penalized.[11]

Our approach to the fee-splitting attempts to deal with these disadvantages by approaching the division rule as an optimization problem. Starting with the same information as de Clippel, et al., we select the allocation that minimizes the discrepancy between it and the firm's reports. We do this in two ways: as a pure optimization problem and then as a statistical model seeking best estimates. By framing the problem in this way, our methods become far more tolerant of incomplete information. We can also be very clear about the connection between our solution and the reports provided by the firms. We can see how the firms interact in their assessments of each other quite explicitly. We think these explicit connections will help convince firms to employ our techniques.

What do we lose in treating our division problem as an optimization or statistical problem? One way to examine this is to consider which of de Clippel, et al.'s axioms

[10] Consider the pair AB. Neither A's nor B's ratings will contain the pair AB, because they are not allowed to self-report. Therefore C's and D's ratings must include A and B to satisfy the two rater requirement.

[11] We should note that there are technical choices in the de Clippel rule that allow one to minimize the impact of collusion. (See Tideman and Plassmann, 2008, p.31). Such choices are buried enough that it would be hard to make transparent which firms are contributing to the problem.

our method satisfies. As will become clear in subsequent sections our methods satisfy both the Objectivity and Consensualness axioms. Our methods are also always efficient in that all of the good is allocated.

The one axiom that fails for our method is Strategy-Proofness. A firm's allocation may depend on its reports about other firms. However, as will become evident in the next section, the connection between the allocation and the reports is attenuated, and nearly impossible to predict a priori. Thus, we believe that it is impractical if not literally impossible for any firm to benefit itself unilaterally by misreporting. Furthermore, both the optimization and statistical approaches turn out to be inherently collusion resistant, because the dense "web" of relative assessments prevent any one or two sets of scores from affecting the outcome by much. We also propose additional methods designed to resist collusion.

2.2 Problem Specification

Suppose that N firms, labeled 1, 2, ..., N, have worked together on a litigation matter. A court has issued a collective award of attorneys fees, and the objective is to distribute the sum according to "desert" as defined by the firms. To begin, we ask each firm to rate the relative contribution made by each of the other firms. We label these observed ratings as S_{ij}, $0 < S_{ij} < 1$ where i is the rater, and j is the rated firm. To prevent self-dealing, a firm may not rate itself. In addition, because some firms may not have sufficient contact with some of the other firms to make an educated rating, some of the S_{ij} may be missing. So for example, we might have a score matrix that looks like Table 2.1.

Rater	Rating for Firm 1	Firm 2	Firm 3	Firm 4
Firm 1	NA	0.50	0.40	0.10
Firm 2	0.70	NA	0.30	NA
Firm 3	0.85	NA	NA	0.15
Firm 4	NA	0.50	0.50	NA

Table 2.1: Example Score Matrix

Note that under this construction, the row sums ($\sum_j S_{ij}$) necessarily equal 1, since each rating firm makes *relative* assessments among the firms for which it has information. Missing values are NA, as distinct from a zero contribution, which are not permitted in most of the models that follow. Given this dataset, the goal of the approaches below is to arrive at a justified estimate for the contribution made by each firm to the litigation as a whole. We will denote by α_j the final recommended allocation for firm j.[12]

2.3 Optimization

Our first approach to the fee allocation problem is optimization. We will choose allocations $\{\alpha_j\}$ so as to minimize the error between our allocations and the reports provided by the firms themselves. We will analyze two different measures. The first is based on pair-wise error and the second is based on the individual error of the firms.

[12]Further note that even a simple example like Table 2.1 already violates the de Clippel informational requirements. For example, Firms 1 and 2 are never rated together.

2.3.1 Pairwise Error

Suppose that firm i reports on the relative contributions S_{ij} and S_{ik} of the two firms j and k. Then we would want our final allocations α_j and α_k to satisfy

$$\frac{S_{ij}}{S_{ik}} \approx \frac{\alpha_j}{\alpha_k}.$$

A natural measure of the error produced by our allocation relative to firm i's assessment of firms j and k is

$$\left(\frac{S_{ij}}{S_{ik}} - \frac{\alpha_j}{\alpha_k}\right)^2$$

but this has a rather severe drawback — it is not symmetric in the firm labels j and k. Alternatively we could choose the symmetric measure

$$\left(\alpha_k\, S_{ij} - \alpha_j\, S_{ik}\right)^2.$$

This formulation has two advantages. It is symmetric in the labels and it leads to a well-behaved quadratic optimization function. Algebraically we have

$$\left(\alpha_k\, S_{ij} - \alpha_j\, S_{ik}\right)^2 = \left(\alpha_k\, S_{ik}\right)^2 \left(\frac{S_{ij}}{S_{ik}} - \frac{\alpha_j}{\alpha_k}\right)^2.$$

So this latter measure amounts to a weighted version of our non-symmetric measure. This weighting will magnify the error when both the report S_{ik} and the allocated amount α_k is large. The former may happen because the reporting firm only has information on a few other firms, thus the relative values will tend to be larger, or because this particular firm merited a large allocation. In either case, we think large reports should be held to stricter requirements forcing the error they produce to be less.

Summing over all firms and their reports we are left the following quadratic optimization problem:

$$\text{Minimize} \sum_{i=1}^{N} \sum_{\substack{j,k \neq i \\ S_{ij}, S_{ik} \neq 0}} (\alpha_k \, S_{ij} - \alpha_j \, S_{ik})^2$$

$$\text{subject to } \sum_i \alpha_i = 1$$

$$\alpha_i \geq 0 \; \forall i \in \{1, 2, \ldots, N\}$$

2.3.2 Individual Error

Another approach would be to focus on the individual assessments of the firms. Suppose firm i reports a relative share S_{ij} for firm j. Then we would want

$$S_{ij} \approx \frac{\alpha_j}{\sum_{\{k|S_{ik} \neq 0\}} \alpha_k}.$$

Reasoning as before we might wish to quantify the error in this allocation as

$$\left(S_{ij} - \frac{\alpha_j}{\sum_{\{k|S_{ik} \neq 0\}} \alpha_k} \right)^2.$$

Unlike the earlier measure, this one exhibits no particular asymmetry. It is, however, rather ill-behaved for optimization purposes. It also has the property of weighting all errors equally, even if the actual allocations involved are rather small. We can create both an easier optimization problem as well as a more meaningful weighting by using

$$\left(S_{ij}\sum_{\{k|S_{ik}\neq 0\}}\alpha_k-\alpha_j\right)^2=\left(\sum_{\{k|S_{ik}\neq 0\}}\alpha_k\right)^2\left(S_{ij}-\frac{\alpha_j}{\sum_{\{k|S_{ik}\neq 0\}}\alpha_k}\right)^2$$

as our quantification of the error in the allocation for firm i's estimate for the relative share for firm j. Note that now we are weighting the error by the amount of the allocation that firm i observes. This leads us to the well-behaved quadratic optimization problem

$$\text{Minimize}\sum_{i=1}^{N}\sum_{\substack{j\neq i\\ S_{ij}\neq 0}}\left(S_{ij}\sum_{\{k|S_{ik}\neq 0\}}\alpha_k-\alpha_j\right)^2$$

$$\text{subject to }\sum_i \alpha_i=1$$

$$\alpha_i\geq 0\ \forall i\in\{1,2,\ldots,N\}$$

2.4 Bayesian Model

An alternative approach to the problem is to use a statistical model in which each firm's "true" contribution to the litigation is modelled as a latent vairable, α_j. Contributions are measured on a relative basis, so the vector $\boldsymbol{\alpha}$ is a unit simplex, that is $\alpha_j > 0$, and $\sum_j \alpha_j = 1$. The α_j's are latent and therefore not directly observed. Instead, we observe the relative ratings given by other firms, which provide information about the α_j's indirectly and with error. In what follows, we develop Bayesian models of increasing sophistication to estimate α_j.

2.4.1 Linear Approach

One straightforward approach is to view the observed scores (S_{ij}) as consisting of the "true" contribution of firm j relative to all of the other firms rated by rater i plus some Gaussian error. So, for example, if we let i be the rating firm, j be the rated firm, and X_{ik} be an indicator variable for when firm i has sufficient contact with firm k to evaluate its contribution, then the observed score can be modelled as:

$$S_{ij} = \mu_{ij} + \epsilon_{ij} \tag{2.1}$$

where $\mu_{ij} = \frac{\alpha_j}{\sum_{k \neq i} \alpha_k X_{ik}}$, and $\epsilon_{ij} \sim N(0, \sigma_j^2)$. Here we give the α_j's a flat Dirichlet prior, since they constitute a simplex, and σ_j's receive uninformative inverse-gamma priors. (Gelman, 2006).

The model in Equation 2.1 can be extended to account for idiosyncratic raters. For example, to the extent that certain raters produce "noisier" signals than others, we can take a random effect approach, where the variance of the error term depends on the rater, i.e., σ_i. Using a random effect may also have the benefit of making the model resistant to strategic behavior, since the model will characterize a rater whose ratings significantly deviate from the norm as "noisy," and thus an aberrant rater's scores will be effectively downweighted. We will explore these ideas further in the compositional model below.

2.4.2 Compositional Approach

A significant limitation to the linear approach is that it fails to account for correlation among the observation errors. Because $\boldsymbol{\alpha}$ is a simplex, error in measuring one firm's contribution should negatively correlate with the error in measuring other firms' contributions. For example, suppose Firm A rates the relative contributions of Firms B & C. If A underestimates B's contribution, then it must necessarily overestimate C's contribution, since the total relative contributions of B & C must sum to 1.

The log-ratio approach to compositional data proposed in Aitchison (1986) helps model this correlated error. Let's assume that we have N firms involved in the division, and that the true (latent) contribution for each firm is represented by α_j, where $j = 1, \ldots N$. Because the vector $\boldsymbol{\alpha}$ is a simplex, it is completely defined by its first $n = N - 1$ elements.[13] So instead of focusing on the individual α_j's, we can focus on their log-ratios, namely:

$$\mu_j = \log\left(\frac{\alpha_j}{\alpha_N}\right), \qquad j = 1, \ldots, n.$$

We can then view the observed log-ratios to be the true log-ratios plus some error term. Let s_{ij} be the observed contribution of firm j as judged by firm i, and the vector $\boldsymbol{s_i}$ contain all of the contributions observed by firm i except the last term (s_{iN}), namely $(s_{ij}, j = 1, \ldots, n)$. Then, we can model those contributions as:

[13] $\alpha_N = 1 - \alpha_1 - \alpha_2 \ldots - \alpha_n$. As Aitchison (1986) shows, the results are invariant to which α_j is chosen to be the α_N – in other words, the results are invariant to permutations of the firms.

$$\log\left(\frac{\boldsymbol{s_i}}{s_{iN}}\right) = \boldsymbol{\mu} + \boldsymbol{\epsilon_i}, \tag{2.2}$$

where $\epsilon_i \sim \mathcal{N}^n(0, \Sigma)$,and i indexes the firm doing the judging. Here, $\alpha_j, j = 1, \ldots, n$, can have uniform priors on [0,1], or perhaps a weakly informative normal prior (with appropriate constraints) since we know that the relative contributions will tend to be in the lower part of that interval.

For ease of notation, we can borrow the notation from Aitchison (1986) and express the model as $\boldsymbol{s_i} \sim \mathcal{L}^n(\mu, \Sigma)$. The covariance matrix (Σ) captures interdependencies among the firm contributions. For example, if Firm A and Firm B worked on the same aspect of a case, then we should expect negative covariance between the error associated with A's contribution and B's contribution.

One final complication to the log-ratio approach is the issue of missing data. Recall that to prevent self-dealing, Section 2.2 specified that firms may not rate themselves. In addition, some firms may not have sufficient information to rate all of the other firms involved in the litigation. The estimated contributions provided by the raters are therefore only the *relative* contributions of the firms for which the rater has enough information. The approach presented thus far, however, requires complete information.

Fortunately, because the logratio approach is based on a multivariate normal model, it can handle relative contributions (known as subcompositions) with ease through a linear transformation. (Aitchison, 1986). If $\boldsymbol{s_i} \sim \mathcal{L}^n(\mu, \Sigma)$ is a complete composition as seen in Equation 2.2, then subcomposition $\tilde{\boldsymbol{s}}_{\boldsymbol{i}}$ has the following distribution:

$$\tilde{\boldsymbol{s}}_{\boldsymbol{i}} \sim \mathcal{L}^n(\tilde{\mu}, \tilde{\Sigma})$$

$$\tilde{\mu} = Q\mu$$

$$\tilde{\Sigma} = Q\Sigma Q^T,$$

where we construct matrix Q based on which indicies are chosen for the subcomposition. In particular, let M be the number of firms in the subcomposition with $m = M-1$, and N be the number of firms in the complete composition with $n = N-1$. Then

$$Q = F_M Z F_N^{\intercal} H^{-1},$$

where F_k is the identity matrix I_k with an appended last column of -1's (i.e., $[I_{k-1} : -j_k]$), Z is the $M \times N$ selection matrix that creates the subcomposition, and H is defined as $H = I + J$, where I is the identity matrix and J is a matrix of ones. (Aitchison, 1986).

We apply this property frequently throughout the remainder of the chapter to do estimates using the compositional model. For brevity, we will assume the use of this transformation to handle relative contributions whenever necessary without explicitly mentioning it.

2.4.3 Collusion Resistance

A limitation of the basic compositional model is that it treats the rater firms as interchangeable scientific instruments, so that each set of observed ratings is like any other. However, certain judging firms may be more knowledgable or competent, and thus they may have lower error in estimating contributions. Of even greater concern, two firms may attempt to collude with each other — for example, they may agree to rate each other at levels far in excess of their desert.

One way to address these concerns is to introduce a random effect into the compositional model to account for rater variability. To do this, we can use a decomposition of the covariance matrix:

$$\Sigma = \Omega D \Omega^T,$$

where D is a diagonal matrix of eigenvalues, σ_j, corresponding to the "scale" of the covariance. We then assume that D has takes the form:

$$D = \gamma_i \tilde{D},$$

where $\tilde{D}$ is a diagonal matrix with elements $\tilde{\sigma_j}$, $0 \leq \tilde{\sigma_j} \leq 1$, and γ_i is the random effect measuring the variability (or reliability) of the rating firm i. We assume that the γ_i arise from a common normal distribution with zero mean and common variance.

The random-effects model directly addresses accuracy differences among different firms. It also helps resist collusion. Consider this example: suppose that there are 10 firms in the pool, and Firm 4 and Firm 5 collude to give higher ratings to each other.

Since Firm 4's estimate of Firm 5's contribution will deviate from the assessments made by the other eight firms, the model will estimate γ_4 to be quite high, effectively downweighting Firm 4's observations. A similar downweighting will occur due to Firm 5's estimate of Firm 4's contributions. Such collusion protection is less effective when the number of firms is small, because detecting "outlier" ratings will be difficult if not impossible. But the random effect model provides some incentive and assurance against collusive behavior when there is a substantial number of firms.

2.5 Examples of Implementation

Simulation 1. The first simulation uses the example data introduced in Section 2.2 and reproduced in Table 2.2. The authors constructed the dataset by taking the ground truth (obviously not observed by the model) and then playing the role of each of the rating firms and doing rough estimates of the other firms. So, for example, Firm 2 only rates Firm 1 and Firm 3, whose contributions in truth should be in a 5:2 ratio, but which we made roughly 70/30. As seen in Table 2.2, the models do a reasonable job estimating the "true" contribution values.[14] The compositional model is notably less accurate, and we suspect this is because of the limited data available (and relatively large number of parameters) in a four firm problem. The posterior distributions of the estimates for the compositional model exhibit a lot of variance, probably due to overfitting. For example, as seen in Table 2.3, the credibility

[14] The Bayesian models were estimated using MCMC methods using the Stan statistical modeling platform. Visual checks of the trace plots suggest that the linear model has difficulty mixing, likely because of the (unmodeled) correlated errors. The compositional model exhibits no such issues.

intervals for the compositional model's estimates are much wider that those for the linear model. These issues (and the relative success of the other methods) suggest that future work might consider some kind of regularization for the compositional model when dealing with small numbers of firms.

	Rating for			
Rater	Firm 1	Firm 2	Firm 3	Firm 4
Firm 1	NA	0.50	0.40	0.10
Firm 2	0.70	NA	0.30	NA
Firm 3	0.85	NA	NA	0.15
Firm 4	NA	0.5	0.5	NA
Truth	0.50	0.25	0.20	0.05
Opt Pair	0.48	0.24	0.21	0.07
Opt Indiv	0.48	0.24	0.21	0.07
Bayes Linear	0.49	0.21	0.21	0.09
Bayes Comp	0.41	0.28	0.24	0.06

Table 2.2: Data and Results for Simulation 1

	Credibility Interval for			
	Firm 1	Firm 2	Firm 3	Firm 4
Linear (50% CI)	(0.49, 0.49)	(0.21, 0.21)	(0.21, 0.21)	(0.09, 0.09)
Linear (95% CI)	(0.41, 0.50)	(0.20, 0.26)	(0.20, 0.26)	(0.07, 0.09)
Comp (50% CI)	(0.22, 0.52)	(0.15, 0.42)	(0.14, 0.31)	(0.04, 0.08)
Comp (95% CI)	(0.01, 0,89)	(0.01, 0.85)	(0.01, 0.66)	(0.01, 0.17)

Table 2.3: Credibility Intervals for Bayesian Estimates in Simulation 1

Simulation 2. The second simulation uses the data in Table 2.4. The dataset was constructed similarly to Simulation 1, except that it involves a larger group of firms and therefore more ratings. We also informally increased the amount of measurement error and introduced more missing data values. With six firms, all of the models perform well at recovering the "truth." Once again, the credibility intervals for the Bayesian compositional model's estimates are much wider that those for the Bayesian linear model, as seen in Table 2.5,[15] and visual checks of the trace plots yielded results similar to Simulation 1.

	Rating for					
Rater	Firm 1	Firm 2	Firm 3	Firm 4	Firm 5	Firm 6
Firm 1	NA	0.60	0.30	NA	0.10	NA
Firm 2	0.50	NA	0.20	0.30	NA	NA
Firm 3	NA	NA	NA	0.50	0.25	0.25
Firm 4	NA	0.45	0.30	NA	0.15	0.10
Firm 5	0.50	0.50	NA	NA	NA	NA
Firm 6	0.60	NA	NA	0.35	0.05	NA
Truth	0.30	0.30	0.15	0.15	0.05	0.05
Opt Pair	0.29	0.27	0.14	0.17	0.06	0.07
Opt Indiv	0.29	0.27	0.14	0.17	0.06	0.07
Bayes Linear	0.28	0.28	0.11	0.17	0.08	0.08
Bayes Comp	0.32	0.28	0.14	0.15	0.05	0.05

Table 2.4: Data and Results for Simulation 2

	Credibility Interval for					
	Firm 1	Firm 2	Firm 3	Firm 4	Firm 5	Firm 6
Linear (50% CI)	(0.28, 0.30)	(0.28, 0.30)	(0.11, 0.15)	(0.15, 0.17)	(0.05, 0.08)	(0.06, 0.08)
Linear (95% CI)	(0.24, 0.32)	(0.25, 0.35)	(0.11, 0.18)	(0.10, 0.19)	(0.02, 0.08)	(0.05, 0.14)
Comp (50% CI)	(0.22, 0.41)	(0.18, 0.32)	(0.10, 0.18)	(0.10, 0.19)	(0.04, 0.07)	(0.04, 0.06)
Comp (95% CI)	(0.04, 0.71)	(0.05, 0.58)	(0.10, 0.40)	(0.10, 0.40)	(0.01, 0.12)	(0.01, 0.10)

Table 2.5: Credibility Intervals for Bayesian Estimates in Simulation 2

Simulation 3. The problem with manual encoding is that measurement errors are introduced in a haphazard and uncontrolled way. Thus, the "truth" is no longer really the truth, since our manual coding may inadvertently bias things in some direction. For the third simulation, we adopted a more systematic approach: We began with the "true" distribution among 10 firms. Then for each rater, we added

[15]Hereafter, we will forego reporting credibility intervals in the interests of brevity.

independent normal error to each component, where the standard deviation of the error was set at 20% of the true value (to model the fact that people are perhaps less precise when observing larger quantities). We then randomly dropped some of these components, although to make things more realistic, more "involved" firms (those entitled to a greater share) were more likely to observe a greater number of their peers. Finally, we renormalized the observations so that the relative shares all added to one. The resulting set of observations was labelled the "Baseline" set for the simulation.

To simulate collusive behavior, we then assumed that Firms 4 and 5 agreed to inflate each other's scores. Table 2.6 shows an example of the baseline (no collusion) set of observations for Firms 4 and 5, and then the affected (collusion) set for the same two firms. Note that as a result of the collusion, Firm 4's rating for Firm 5 is about twice as large, whereas Firm 5, who would have not rated Firm 4 at all, instead rates Firm 4 greater than Firm 1, the greatest contributor in the set. Those inflated scores then have concomitant downstream effects on the other firms rated.

	Rating for Firm									
	1	2	3	4	5	6	7	8	9	10
Baseline 4	0.42	NA	0.28	NA	**0.19**	NA	0.11	NA	NA	NA
Baseline 5	0.36	0.26	0.21	**NA**	NA	0.17	NA	NA	0.079	0.083
Collusion 4	0.32	NA	0.21	NA	**0.39**	NA	0.08	NA	NA	NA
Collusion 5	0.21	0.16	0.13	**0.39**	NA	0.10	NA	NA	NA	NA

Table 2.6: Baseline and Collusion Data for Simulation 3

Table 2.7 displays the true contribution for each of the ten firms in Simulation 3, and the results of the various models on the Baseline and Collusion datasets. Three results are especially worthy of note. First, the outputs of the four models are again quite similar. Second, the models seem to have some natural resistance to collusion, probably because they rely on the observations of so many other firms. Despite Firm 4 and Firm 5's significant attempts to inflate their scores, they are at best only able to modestly affect outcomes by a few percentage points, which is often within the general noise we see in the table. Third, the Bayesian collusion resistant model seems able (at least in this example) to negate the distortion created by Firm 5's collusive efforts.

	Allocation									
	1	2	3	4	5	6	7	8	9	10
Truth	0.25	0.15	0.15	0.10	0.10	0.10	0.05	0.05	0.025	0.025
Baseline Set										
Opt Pair	0.24	0.16	0.14	0.11	0.07	0.12	0.05	0.06	0.032	0.026
Opt Indiv	0.24	0.16	0.13	0.12	0.08	0.12	0.05	0.06	0.029	0.024
Bayes No Resist	0.25	0.15	0.13	0.11	0.09	0.11	0.05	0.06	0.028	0.028
Bayes Resist	0.23	0.16	0.13	0.11	0.09	0.10	0.05	0.06	0.032	0.029
Collusion Set										
Opt Pair	0.20	0.14	0.11	**0.14**	**0.09**	0.11	0.05	0.08	0.058	0.043
Opt Individual	0.20	0.15	0.11	**0.16**	**0.09**	0.10	0.04	0.07	0.045	0.036
Bayes No Resist	0.24	0.15	0.13	**0.13**	**0.08**	0.10	0.05	0.06	0.028	0.028
Bayes Resist	0.25	0.14	0.13	**0.11**	**0.07**	0.10	0.05	0.06	0.027	0.027

Table 2.7: Results for Simulation 3

One should bear in mind that all of these results are subject to random variation. We construct the baseline dataset through random processes, and the process of constructing the collusion set (which attempts to distort the baseline set) also has random aspects. Further, the estimation procedure for the Bayesian model, which uses Markov Chain Monte Carlo (MCMC) methods, also has random aspects. To get a better sense of average model performance for the Bayesian models, we ran a procedure similar to Simulation 3 (dataset generation as well as model estimation) twenty times, measuring the performance of the models by calculating the sum of square errors between the estimated and true parameters, namely:

$$\text{Performance Metric} = \sum_j (\hat{\alpha_j} - \alpha_j)^2$$

The results of this exercise are seen in Table 2.8. It shows that the non-collusion-resistant model predictably performs less well on the collusion dataset than on the baseline dataset. This result is of course expected, since the colluders are distorting the observations. The collusion-resistant model, however, seems able to counteract some of the collusion, getting on average closer to the "truth" than the non-resistant model.

	Dataset / Bayesian Model		
	Baseline / No Resist	Collusion / No Resist	Collusion / Resist
Mean Performance	1.49E-3	2.65E-3	1.98E-3

Table 2.8: Average Mean Squared Error from 20 Runs of Simulation 3

2.6 Discussion

Both the optimization and statistical methods offer more viable and flexible methods for solving the fee division problem. They are more easily understood and interpreted: The optimization method seeks a compromise that minimizes the error between it and the firm reports. The statistical method determines the most likely underlying "true" allocation assuming that the firm reports are noisy estimates of this truth.

The proposed methods are also capable to handling incomplete data and are robust to collusion, with the collusion resistant Bayesian model being particularly so. In addition, in simulations, the methods consistently yield allocations that accord with each other and that are close to the ground truth. The close agreement between the optimization and statistical models may seem remarkable at first, but further consideration may suggest that it should be unsurprising. As we know from the Gauss-Markov Theorem in the linear regression context, the least squares estimator (optimization approach) is the best linear unbiased estimator (statistical approach). Perhaps what we see here is a compositional data variant of the Gauss-Markov Theorem, although further research would be needed to ascertain which precise methods are linked and under what conditions.

2.6.1 Self-Reporting

One potentially significant limitation of our proposal is its prohibition on self-reports — firms rating themselves. This limitation raises two concerns. First, firms may object to these procedures because they cannot provide any direct input on the share they receive. Part of this objection may rest on some kind of participation value, but it may be also related to information costs. Arguably, the firm with the best (although not unbiased) information on what Firm A did during a litigation is Firm A itself. By not allowing self-reporting, we may be throwing away valuable information. Second, if firms are unable to self-report, what incentive do they have to make careful assessments about other firms? When a firm is self-reporting, it is highly motivated to get other firm contributions right, because its own contribution will be assessed relative to those other contributions. When a firm cannot self-report, the relationship between its assessment of other firms and what the rater firm ultimately receives may appear much more tenuous.

The good news is that our methods are actually capable of handling self-reporting.[16] Just as the methods exhibit natural resistance to collusion, they also resist distortions caused by overly generous self-reports. Much of this resistance likely comes from the dense, interlocking web of relative ratings that effectively prevents any one set of ratings from distorting the outcomes. In the case of the collusion resistance model, there are further helpful incentives at play. Due to the presence of the random effect term, a self-report that significantly departs from the underlying consensus effectively causes the model to "discount" that rater firm's report. Thus, if a firm wishes its report to be taken seriously, it would be wise to rate itself and the other firms carefully and accurately.

To test how our methods handled self-reports, we extended the simulation in Table 2.7 to have self-reports. We estimated the allocations for four sets of data, all based on the original baseline set from Simulation 3: i) the baseline set as before; ii) the baseline set plus unbiased self-reports; iii) the baseline set with self-reports where Firm 3 doubled its own rating; iv) the baseline set with self-reports where Firm 9 quadrupled its own rating. As seen in Table 2.9, the models appear unfazed by the addition of self-reports, whether unbiased or distortive.

[16]Our thanks to Ben Alarie for pointing out this possibility.

	Rating for Firm									
	1	2	3	4	5	6	7	8	9	10
Truth	0.25	0.15	0.15	0.10	0.10	0.10	0.05	0.05	0.025	0.025
Opt Pair										
No Self-Report	0.24	0.16	0.14	0.11	0.07	0.12	0.05	0.06	0.032	0.026
Unbiased Self-Report	0.23	0.17	0.14	0.11	0.08	0.11	0.05	0.06	0.024	0.025
Firm 3 Inflated Self	0.23	0.16	0.16	0.11	0.08	0.11	0.05	0.06	0.026	0.027
Firm 9 Inflated Self	0.22	0.16	0.13	0.11	0.09	0.11	0.05	0.06	0.060	0.024
Opt Indiv										
No Self-Report	0.24	0.16	0.13	0.12	0.08	0.12	0.05	0.06	0.029	0.024
Unbiased Self-Report	0.23	0.16	0.14	0.12	0.08	0.11	0.05	0.06	0.025	0.025
Firm 3 Inflated Self	0.22	0.16	0.16	0.11	0.08	0.11	0.05	0.06	0.028	0.029
Firm 9 Inflated Self	0.22	0.16	0.13	0.11	0.09	0.11	0.05	0.06	0.053	0.023
Bayes No Resist										
No Self-Report	0.25	0.15	0.13	0.11	0.09	0.11	0.05	0.06	0.028	0.028
Unbiased Self-Report	0.24	0.16	0.13	0.12	0.08	0.12	0.05	0.06	0.027	0.027
Firm 3 Inflated Self	0.23	0.15	0.14	0.10	0.07	0.12	0.05	0.06	0.027	0.027
Firm 9 Inflated Self	0.22	0.16	0.13	0.11	0.07	0.12	0.05	0.06	0.030	0.028
Bayes Resist										
No Self-Report	0.23	0.16	0.13	0.11	0.09	0.10	0.05	0.06	0.032	0.029
Unbiased Self-Report	0.22	0.16	0.13	0.10	0.09	0.12	0.05	0.06	0.025	0.025
Firm 3 Inflated Self	0.26	0.17	0.13	0.10	0.06	0.12	0.05	0.06	0.025	0.025
Firm 9 Inflated Self	0.22	0.16	0.13	0.10	0.09	0.12	0.05	0.06	0.032	0.028

Table 2.9: Results for Self-Report Simulation

2.6.2 Data Concerns

Our proposal has significantly more relaxed data requirements than the de Clippel rule, since it merely requires that every firm be rated by some other firm, not that every pair of firms be rated by some other firm. However, for the methods to be effective, not only do the firms have to be rated, but there must be some degree of interconnection between the various ratings. It is unlikely that any firm will have no ratings, but small-share firms may have only one point of connection to the broader group. For example, A, B, and C all work together, but D only interacts with C. Such small-share firms are somewhat vulnerable to the whims of their connection point.

Another structure that could be troubling is a dumbbell-shaped social network where one firm is the link between two groups of firms (ie., a low-degree vertex with high betweenness). In this situation, the relative weights within the groups may be well estimated, but the weight between the two groups will depend entirely on the link firm. Whether this dumbbell shape creates problems in practice though is unclear. After all, one suspects that how satisfied a firm is with its allocation depends on how its portion compares to firms with which it is familiar, not portions received in the foreign half of the dumbbell.

Another potential concern is that ratings will be correlated with share size. This bias can occur in two ways. One possibility is that prominence can psychologically affect the perception of a firm's contribution. The other is that people may not assess small differences in proportions very well – for example, the difference between 33% and 50% is better understood than the difference between 3.3% and 5%. The extent that these biases are true is an empirical question and will require additional investigation. For now, we only argue that the direction of the bias is completely unclear ex ante. A rater may perceive a large-share firm as being overly important because of its prominence, or a rater may perceive a small-share firm as being overly important because of a bias toward awarding equal shares to each entity. Similarly, any round-off error associated with small values presumably cuts in both directions. And round-off error can be reduced in litigations featuring a large number of firms by simply reducing the maximum number of firms than any rater can assess (which will have the effect of increasing the relative proportions involved).

A system based on peer ratings may also be susceptible to preening as well as racial or gender bias concerns. Since allocations are based on peer perception, firms may waste resources trying to look good. Similarly, peer ratings may incorporate unstated biases against women or minority attorneys. We concede that these are potential problems with any system reliant on peer ratings, but nothing in our proposal exacerbates them. The analog to preening in the lodestar context is running up billable hours, a far more effective and predictable way of increasing one's share of the fees than preening. As for discrimination, the weights used in the lodestar method is also a way in which bias can creep into its "objective" calculus. And at least for our statistical method, in cases with a large number of firms, it may be possible to use covariates to check for possible race effects both for raters and ratees.

Finally, we note that nothing in our proposal requires that the raters in our scheme be identical to the rated firms. So a court could easily add to the dataset its own assessment of the firms (or the assessments of consultants or special masters) to address any of the above concerns. For reasons stated in the Introduction, we doubt that courts or special masters will often have such superior information so as to justify this strategy. Also, a single set of ratings is unlikely to influence the outcome, for the same reasons that make our methods collusion resistant. Nonetheless, courts can use this option as a safety valve, especially if they weight the "neutral" ratings more heavily in the model.

3 Latent Space Models for Legal Doctrine

Legal researchers frequently use a case's doctrinal area or subject matter as a standard covariate or predictor. For example, the well-known Supreme Court Database divides the United States Supreme Court's cases into fourteen major subject areas, such as First Amendment, Criminal Procedure, and Judicial Power. The database then breaks down these major areas into smaller issue areas based on the legal issues presented. (Spaeth et al., 2013). Subject matter can be useful to researchers because of its correlation with judicial attitudes, precedential weight, case complexity, and a whole host of other attributes that may be relevant to a given study. Conventionally, however, a case's subject matter has been manually coded, leading to concerns about subjectivity, arbitrariness, and confirmation bias in the coding. (Harvey and Woodruff, 2011, 423)

To address these concerns, this Chapter proposes using case citations to estimate the "location" of cases in a latent subject matter space as an alternative to manual coding. Then, using a dataset of First Amendment cases, we show that the citation-derived latent space model produces a usable map of First Amendment doctrine. The map correlates well with what legal actors might expect, captures crossovers between doctrinal areas, and avoids the subjectivity of manual coding.

3.1 Motivation and Literature Review

As Harvey and Woodruff (2011) and Sherry (2004) note, the manual classification of legal cases into discrete doctrinal categories can be an artificial and fraught exercise.

(Harvey and Woodruff, 2011; Sherry, 2004). Because manual classification is subjective, resesarchers will often disagree about the proper category. In addition, because cases cut across traditional doctrinal categories, selecting a single classification can arbitrary, oversimplistic, and possibly distorting. Take, for example, *City of Erie vs. Pap's A.M.*, a Supreme Court case involving a local public indecency ordinance and an adult entertainment club. Within First Amendment doctrine, does one classify *City of Erie* as an obscenity case or one about commercial speech? To the extent that the city was concerned about "secondary effects" — crimes associated with the presence of adult establishments — should *City of Erie* be classified as a "forum doctrine" case as well?

Worse yet, Harvey and Woodruff (2011) has suggested that the manual coding of issue areas in the widely used Supreme Court Database, (Spaeth et al., 2013), may be afflicted by confirmation bias. Specifically, Harvey and Woodruff (2011) found that for cases that implicated several plausible issue areas, "the assignment of issue codes . . . [was] conditional on both case disposition and the known preferences of the deciding court." Understandably, how we subjectively classify a case may depend on its outcome, but that kind of confirmation bias makes existing datasets problematic for models attempting to predict or explain legal outcomes.

To address these coding issues, we propose using citation networks as the means for characterizing cases. Citations, a familiar mainstay of traditional legal argument, have drawn recent interest among researchers interested in social network theory. For example, Fowler et al. (2007) and Fowler and Jeon (2008) collect all citations among United States Supreme Court cases through 2005 and construct a network to identify the most important precedents over time. Lupu and Fowler (2013) use the Supreme Court case citation network to ascertain whether Supreme Court justices cite precedent strategically. Other scholars have investigated ways for representing the Supreme Court's citation network. (Bommarito II et al., 2009). For example, Rabina and Sula (2015) produces data visualizations of the First Amendment citation network. These prior citation network studies, however, study the networks as networks, and do not involve the use of probability models to measure distance or to classify cases. Dunn and Ruiz (2015) discusses an algorithm for determining whether a case is a "patent case" using the unknown case's citations, but it is restricted to this classification problem and relies on manually tagged opinions.

Perhaps the closest prior work to the models proposed below is Clark and Lauderdale (2010), which uses a latent space model to locate Supreme Court justices and their opinions in a common ideological space. Their efforts, however, treat case citations as a measure of ideological endorsement, and focus on testing theories about Supreme Court bargaining.

3.2 Methods

We build on the latent space approach to social networks first proposed by Hoff et al. (2002). We model all of the cases as existing in an unobserved latent space, which we assume to be two-dimensional here for demonstration purposes. The likelihood of case k citing case l depends on the Euclidean distance between them, and is conditionally independent of all other citations given the cases' positions. Specifically, we assume

$$\text{logit}(q_{kl}) = \alpha - ||x_k - x_l||^2 \tag{3.1}$$

$$C_{kl} \sim \text{Bern}(q_{kl})$$

where q_{kl} is the probability that case k cites case l, C_{kl} is a binary observation representing whether k actually cites l, α is a scaling parameter, and x_k is the latent (two-dimensional) position of case k. Unlike more conventional social networks, case citation networks are unidirectional, because earlier cases cannot cite later ones. Under the proposed model, we treat such "prospective" citations (citations to future cases) as missing values rather than as zero or "non-citation." This coding is important because we are using citations as evidence of subject matter affinity between cases. The failure of an earlier case to cite a later one is not due to lack of affinity but rather lack of opportunity. An alternative coding would be to impose symmetry, so that a citation from a newer case A to an older case B implies a prospective citation from B to A.

Latent space models notoriously have rotational indeterminacy problems, because the likelihood is invariant to a rotation of the entire latent space. This indeterminacy is closely analogous to the "label switching" problem in mixture models, in which the labels for the various elements of the mixtures can be arbitrarily permuted. (Jasra et al., 2005). The latent space estimated here is no different, since the locations of x_k can be rotated with impunity. The sampling estimates must therefore be suitably transformed to yield meaningful results, for which a generalized Procrustes analysis is the conventional approach. (Hoff et al., 2001; Martin and Quinn, 2002; Dryden and Mardia, 1998).

A natural extension of the model is to include random effect terms to capture case-specific attributes and to prevent them from contaminating the latent position estimates. For a variety of reasons, certain cases may be more generous in their citation to other cases, Lupu and Fowler (2013), and that greater propensity to cite should not be conflated with topic affinity. On the flip side, some cases are more likely to be cited for reasons unrelated to their topic affinity – for example, because they are more famous or broader in scope. Adding two sets of random effects controls for these effects:

$$\text{logit}(q_{kl}) = \alpha + \gamma_k + \delta_l - ||x_k - x_l||^2, \tag{3.2}$$

where γ_k is the random effect for the citing case, and δ_l for the cited case.

We can generalize this model beyond a binary regression to consider the strength of the citation link. For example, since "closeness" is likely related to the citation intensity (the proportion of citations that a case occupies), a beta regression might be an option, or one can use a Poisson regression to capture the number of citations to a case. For purposes of this Chapter, however, we will focus on binary regression.

3.2.1 Controlling for Strategic Citation

One important concern for the latent space model is strategic citation, specifically citation based on ideology. Citing past precedent is a form of persuasion, and thus one would expect opinion writers to be more likely to cite cases in support of their outcome. Credibility of course demands that any judicial opinion cite all directly relevant precedent, but at the margin, whether a case is cited or omitted may be closely correlated with outcome. To the extent that we would like the latent citation space to reflect subject matter only, the model needs to control for strategic citation to avoid clusters based on outcome.

A straightforward way to control for strategic citation is to code cases for ideological outcome and use that information as a covariate. For example,

$$\text{logit}(q_{kl}) = \alpha + \gamma_k + \delta_l - ||x_k - x_l||^2 + \beta Z_{kl} \tag{3.3}$$

where Z_{kl} is a binary variable representing whether case k and case l have the same ideological outcome (liberal or conservative). The problem with this approach, however, is that it again involves manual coding and the subjectivity and arbitrariness that entails. Whether an outcome is liberal or conservative is frequently unclear — for example, an opinion promoting the free exercise of religion may be "conservative" because it furthers current Republican political platforms, or it may be "liberal" because it strengthens civil rights. Cases can also have some conservative outcomes and some liberal ones.

To control for strategic citation without manual coding, we propose using justice agreement as a measure of outcome similarity. Justices choose to either join, concur, or dissent with a majority opinion, a relatively more objective observation. (Because recent Supreme Court justices have a tendency to fracture their votes, agreeing to some parts and disagreeing with others, even coding "join" votes involves some subjectivity. Such fracturing is notably less frequent at lower appellate levels, and so this subjectivity will be considerably reduced outside the Supreme Court context.) If a justice joins both case k and case l, that gives an indication that the two cases have similar ideological outcomes. If a justice joins one but dissents on the other, then those cases are likely different, at least in that justice's perspective. By tallying the number of justices who vote similarly versus dissimilarly on two cases, we can get a sense of how close the cases are ideologically.

$$\mathrm{logit}(q_{kl}) = \alpha + \gamma_k + \delta_l - ||x_k - x_l||^2 + \frac{\beta}{J}\sum_j Z_{jkl} \tag{3.4}$$

where j indexes the J justices who participated in both case k and l, and Z_{jkl} captures whether justice j voted in the same way in cases k and l. Using this similarity measure avoids the hazards of ideological coding, and allows the justices, and not some researcher-defined distinction between liberal and conservative, to tell us which outcomes go together. A justice voting the same way in two cases provides evidence that those two cases cohere under some perspective, whether political, philosophical, legal, or otherwise, and to the extent the two cases cohere, they are more likely to cite each other.

A potential problem with the above agreement metric is missing data. Model 3.4 requires the existence of "bridge justices" who participate in both case k and case l. Because Supreme Court justices tend to have long tenures, some case pairs will have such bridge justices, but for cases substantially separated in time, the number of bridge justices will be few if not zero. We can address this problem by modelling the agreement metric through a (one-dimensional) latent space, this time for case ideology, and then using those results as a covariate in the citation space model.

$$Z_{kl} \sim \text{Bern}(\theta_{kl})$$

$$\theta_{kl} = \alpha_a - ||y_k - y_l||^2 \tag{5a}$$

where for any pair of cases k and l, the probability of having a justice vote the same way on both cases is θ_{kl}, our measure of ideological similarity. Observed agreements or disagreements (of which there may be up to nine), Z_{kl}, are treated as independent Bernoulli draws with that latent probability. α_a is a scaling parameter, and y_k is case k's position in a latent ideological space. α_a has a flat normal prior. The y_k's have normal priors with some common variance, σ_y, which itself has a flat or weak Cauchy prior. (Gelman, 2006). A Euclidean latent ideological space offers a good conceptual fit because it preserves the transitive properties of a similarity metric. If case k is ideologically similar to case l, and case l is similar to case m, we would expect k to be similar to m. The latent space allows us to borrow strength across cases and impute the agreement metric for time-separated cases.

We can then use the estimates for θ from Equation 5a as covariates in a citation model similar to Model 3.4.

$$C_{kl} \sim \text{Bern}(q_{kl})$$

$$\text{logit}(q_{kl}) = \alpha_c + \gamma_k + \delta_l - ||x_k - x_l||^2 + \beta\hat{\theta_{kl}} \tag{5b}$$

where α_c and β have flat normal priors, the γ_k's have normal priors with mean zero and some common variance σ_γ, and the δ_l's have normal priors with mean zero and some common variance σ_δ. As we are assuming a two-dimensional latent space for this study, the x_k have multivariate normal priors with mean at the origin and a common covariance matrix.

3.2.2 Predicting Future Votes and Cases

In theory, one can also make predictions of future Supreme Court votes by adapting Equations 5a and 5b. Such a model, however, implicitly assumes that justices only vote on the basis of ideology, rather than on "legal" factors such as precedent. In addition, Model 5b is designed to estimate doctrinal space, using judge agreement as a rough measure of ideological similarly, and thus it may be a rather blunt instrument for prediction purposes. Regardless, the predictive power of such an ideology-based model is arguably worth investigating, and the model has the important advantage of making predictions without any need for (or risk of) manual coding.

To predict a case, which we label as $\tilde{k}$, we focus on the lower court opinion, since by definition pending cases have no associated Supreme Court ruling. In keeping with the model's emphasis on "agreement," we estimate the probability that a given justice will "agree" with the lower court opinion, which directly translates into a predicted vote to affirm. The given justice can also "disagree," which naturally translates into a predicted vote to reverse.

The prediction procedure proceeds as follows:

Step 1: Estimate Latent Spaces with Training Set. We estimate Models 5a and 5b using a training dataset of existing Supreme Court opinions, obtaining parameter estimates for α_a, α_c, and β, as well as the latent space locations (x_k, y_k) and random effect terms (γ_k, δ_k) for each case k in the training set.

Step 2: Estimate Location of Test Cases. We then fit the following model using the test data:

$$C_{\tilde{k}l} \sim \text{Bern}(q_{\tilde{k}l})$$

$$\text{logit}(q_{\tilde{k}l}) = \widehat{\alpha_c} + \gamma_{\tilde{k}} + \widehat{\delta_l} - ||x_{\tilde{k}} - \widehat{x_l}||^2 + \hat{\beta}\,\text{logit}^{-1}(\widehat{\alpha_a} - ||y_{\tilde{k}} - \widehat{y_l}||^2) \quad (6)$$

where $\tilde{k} = K+1, \ldots, K+\tilde{K}$ are the testing set cases with lower court opinions, and $l = 1, \ldots, K$ are the training set of Supreme Court opinions. $C_{\tilde{k}l}$ are the citations observed in the lower court opinion of case $\tilde{k}$.

The only parameters to be estimated in Model 6 are the random effect terms $\gamma_{\tilde{k}}$, and the latent space positions, $x_{\tilde{k}}$ and $y_{\tilde{k}}$. The model treats all remaining would-be parameters as fixed, as emphasized by the hats (e.g., $\widehat{x_k}$). These fixed parameters were estimated in Step 1 using the training set of Supreme Court opinions. Fixing them here is conceptually justified because the underlying latent spaces are defined by the training set. All we are doing at the prediction stage is locating the test case $\tilde{k}$ in those latent spaces. Thus, the unknown parameters $\gamma_{\tilde{k}}$, $x_{\tilde{k}}$, and $y_{\tilde{k}}$ are all assumed to arise from the same hyperdistributions previously estimated.

Step 3: Calculate Predicted Votes. Given the estimates found in Step 2, we can calculate the probability that a given justice will vote to affirm the lower court opinion in case $\tilde{k}$. From Equation 5a, we estimate $\theta_{\tilde{k}l}$, the probability that a justice will vote the same way (i.e., agree or disagree with both) in the prediction case $\tilde{k}$ and training case l.

$$\text{logit}(\widehat{\theta_{\tilde{k}l}}) = \widehat{\alpha_a} - ||\widehat{y_{\tilde{k}}} - \widehat{y_l}||^2$$

Next, we use $\widehat{\theta_{\tilde{k}l}}$ along with the justice's known voting record, which we will label V_{jl}, $l = 1, \ldots K$. For ease of notation, we code V_{jl} as 1 when the justice joins the majority opinion, -1 when the justice dissents, and 0 when the justice did not participate.

Assume (without loss of generality) that the justice's vote in $\tilde{k}$ will be to affirm (i.e, $V_{j\tilde{k}} = 1$). Then, the agreement vector, z^A, which describes whether a justice voted the same way in case $\tilde{k}$ and case l contains elements:

$$z^A_{jl} = V_{jl},$$

where z^A is coded 1 for same, -1 for different, and 0 for unknown. We can generate an analogous vector, z^R, under the opposite assumption that the justice's vote in $\tilde{k}$ will be to reverse (i.e., $V_{j\tilde{k}} = -1$). In this latter case,

$$z^R_{jl} = -V_{jl}.$$

Finally, because the justice's voting record V_{jl} is fixed, the only random variable is the vote in $\tilde{k}$, and the only two possible values for the agreement vector $\mathbf{z}$ are z_A

and z_R. Accordingly,

$$P(V_{j\tilde{k}} = 1|\theta_{\tilde{k}\cdot}, V_{jl}) = \frac{p_A}{p_A + p_R}.$$

in which

$$\begin{aligned} p_A &= \prod_l \theta_{\tilde{k}l}^{\frac{1+z_l^A}{2}} (1 - \theta_{\tilde{k}l})^{\frac{1-z_l^A}{2}} \\ &= \prod_l \theta_{\tilde{k}l}^{\frac{1+V_{jl}}{2}} (1 - \theta_{\tilde{k}l})^{\frac{1-V_{jl}}{2}} \end{aligned}$$

and

$$\begin{aligned} p_R &= \prod_l \theta_{\tilde{k}l}^{\frac{1+z_l^R}{2}} (1 - \theta_{\tilde{k}l})^{\frac{1-z_l^R}{2}} \\ &= \prod_l \theta_{\tilde{k}l}^{\frac{1-V_{jl}}{2}} (1 - \theta_{\tilde{k}l})^{\frac{1+V_{jl}}{2}} \end{aligned}$$

Note that given the 1 and -1 coding, the equations above are those typical of a series of independent Bernoulli random variables. Missing values for V_{jl}, which are coded 0, would appear to create spurious product terms but because those spurious terms arise in both p_A and p_R whenever V_{jl} is missing, they cancel out.

Predictions follow directly by using a threshold such as 0.5, but more importantly, this equation generates a probability for an affirm vote, which gives us a measure of confidence for our prediction. Combining the predicted votes for all nine sitting justices generates a prediction for the outcome in case $\tilde{k}$.

3.3 Application and Discussion

The United States Supreme Court provides a natural starting point for any study of legal doctrine. Supreme Court cases have been intensively studied by legal and political scholars alike, and they form something of a baseline in the literature on judicial decision making. As an application of the above latent space models, we thus chose First Amendment cases decided by the Supreme Court between 1946 and 2002. The choice of the First Amendment is largely arbitrary, though it is attractive as a well-defined area in which the Supreme Court regularly hears important cases. We chose to limit the time frame to 1946– 2001 for training purposes and 2002 for test purposes largely for convenience and comparison reasons. The 2002 Term was the year used for the most well-known attempt at Supreme Court prediction. (Ruger et al., 2004).

The data used primarily came from well-known and well-used sources. Citation network data came from the Supreme Court Citation Network dataset developed by Fowler and Jeon (2008). (See also Fowler et al., 2007). Fowler and Jeon (2008) provides a comprehensive list of case citations between Supreme Court majority opinions. Other case specific information such as the justices' votes and case dispositions came from the Supreme Court Database from Spaeth et al. (2013). For prediction modeling, we manually coded the citations of the lower court opinions associated with the 2002 Term cases.

After suitably combining these data sources in R, Models 5a, 5b, and 6 were fit using Markov Chain Monte Carlo methods using the STAN statistical computing platform. (Stan, 2015). Visual checks of the trace plots were performed to ensure proper mixing. The R script and the STAN code are available in the Appendix.

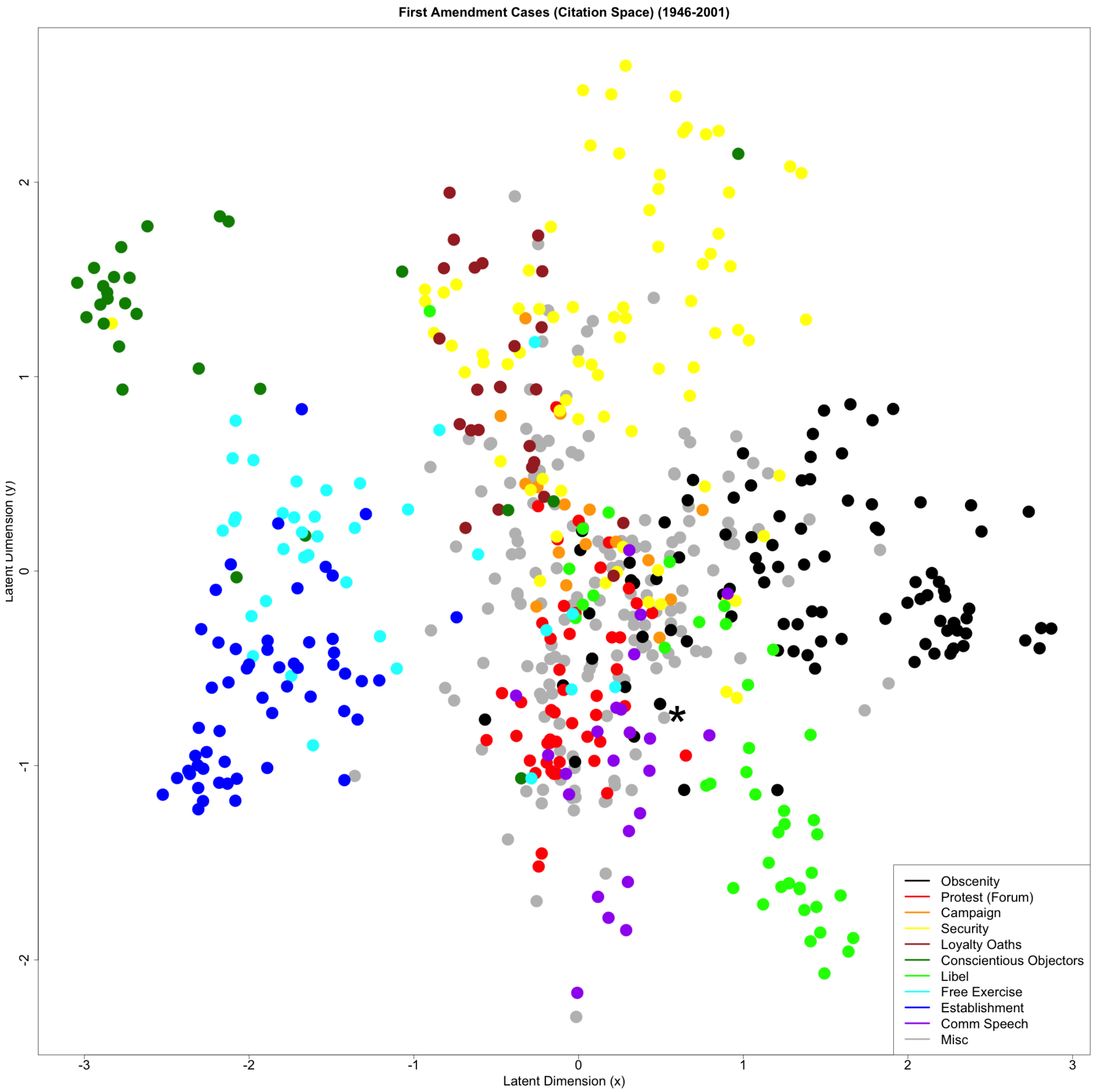

Figure 3.1: Estimated Latent Space for First Amendment Cases

3.3.1 Doctrinal Space

Figure 3.1 displays the result of applying Models 5a and 5b to First Amendment cases from 1946-2001. The colors in Figure 3.1 represent the First Amendment subissue involved in the case as manually coded in the Supreme Court Database. As noted earlier, such manual classifications can raise concerns about oversimplication and bias, but they nevertheless provide a convenient way of validating the model.

Models 5a and 5b are able to separate disparate areas of First Amendment law. Conscientious objector cases occupy their own region in the upper left, whereas areas such as libel, commercial speech, security, and obscenity occupy distinct regions of the latent space. At the same time, the latent space is able to capture "crossover" areas. Many religion cases involve both free exercise and establishment aspects, as evinced by the overlap of the light and dark blue regions. The religion cases as a whole, however, correctly separate from the rest of First Amendment doctrine, which generally deal with speech, not religion. Forum doctrine cases often overlap with other areas of First Amendment doctrine, and thus the forum cases are mixed in with commercial speech, obscenity, and campaign finance.

City of Erie, the adult entertainment case discussed in Section 3.1, is labelled with a star in Figure 3.1. It sits at the intersection of forum doctrine, commercial speech, libel, and obscenity, but contrary to the obscenity classification given to it by a manual coder, the latent space suggests that legally speaking, it is more of a commercial speech case than anything else.

One final advantage seen in Figure 3.1 is found in the gray points, which correspond to cases which manual coders labelled as miscellaneous, a classification with little content. For these cases, the latent space model finds a position in doctrinal space rather than leaving them unclassified.

3.3.2 Prediction

Applying the procedure outlined in Section 3.2.2, we used the 1946-2001 dataset along with citations from the lower court opinions in the 2002 cases to generate predictions for the justices' votes on First Amendment cases for the 2002 Term. The results were a disappointing 30% accuracy rate, a marked contrast from the model's mapping of doctrinal space.[1] The fault for the poor performance does not rest with the use of lower court opinions (which may cite precedent differently from Supreme Court opinions). Running the same procedure using the citations found in the ultimate Supreme Court opinions fares no better.

One possible explanation for the poor predictive performance is that the key parameter estimates may be too noisy for prediction. When mapping doctrinal space, the model aggregates the justices votes to generate a rough estimate of ideological similarity. This similarity measure may be useful for controlling for ideology (a fact further supported by the fact that the estimate for β is relatively large and has a narrow posterior distribution). However, the ideological similarity measure may not contain enough information to help predict the justice's votes. First, the point estimates for the θ_{kl}'s are all quite close to 0.5 (typically ranging from 0.4 to 0.7). Second, the predictions generated by the model frankly lack nuance. The model predicted "affirm" for all of the justices in all cases except for Chief Justice Rehnquist, and Justices Scalia and Thomas, for whom it uniformly predicted "reverse."

[1]By contrast, Ruger et al. (2004) has a roughly 80% accuracy rate in predicting the justices' votes in the 2002 First Amendment cases, whether one uses the outcomes as coded in Ruger et al. (2004) or in Spaeth et al. (2013). To calculate the accuracy of Ruger et al. (2004), one has to exclude Illinois ex rel. Madigan v. Telemarketing Associates, which was a case granted certiorari after the commencement of the Ruger study.

3.3.3 Future Directions

Although Figure 3.1 shows that citations can provide an effective means for locating cases in a latent space, there are some extensions that future work might address. One is to explore other methods for measuring the closeness of cases, such as latent Dirichlet allocation, (Blei et al., 2003), or natural language processing methods, (e.g, Merlo et al., 2013). Although citations provide important information on the relationship among cases, the text of case opinions surely can provide valuable additional data.

Another extension is to improve the predictive performance of the latent space model by drawing on the social recommender literature. (See, e.g., Liu et al., 2013; Gartrell et al., 2012; Ma et al., 2011). Recommender systems have been an area of special interest to online retailers hoping to introduce consumers to new products based on their past purchases or ratings. (E.g., Condliff et al., 1999). With the advent of social media, researchers have incorporated social network data into recommender systems on the expectation that a person's tastes are likely to be similar to his/her friends. In addition to providing another data source, social network data addresses the longstanding issue known as the "cold-start" problem, in which recommenders have trouble with new customers with minimal past history.

Case prediction is deeply analogous to the recommender problem. Just as consumers rate products, judges "rate" legal cases by voting to affirm, concur, or dissent, and we can use the same techniques to help predict what new cases a judge may "like." The case citations are analogous to social network data, albeit as a mirror image. In a conventional social recommender system, the users are networked. In case prediction, the cases (analogous to the consumer products) are networked. The inverted data structure however does not change the model. If our interest is in prediction and not understanding some causal relationship, we can easily pretend that the cases rate the judges, rather than the other way around. Future work might therefore try to apply social recommender models to the case prediction context.

Other Related Work

Below are some shorter published works completed during my program that are not formally part of this dissertation but offered for the Committee's additional consideration.

- Cheng, E.K. (2016) A Bayesian look at the Baby Annie case. *Chance*, **29**, 27–31.
- Cheng, E.K. (2013) Is high-altitude mountaineering Russian roulette? *Journal of Quantitative Analysis of Sports*, **9**, 1–14.
- Cheng, E.K. and Farmer, S.J. (2013) A normalized scoring model for law school competitions. *Green Bag*, **17**, 377–393.

Bibliography

Aitchison, J. (1986) *The statistical analysis of compositional data*. Monographs on statistics and applied probability. Chapman and Hall. URLhttps://books.google.com/books?id=RHKmAAAAIAAJ.

Allapattah Services, Inc. v. Exxon Corp. (2006) 454 F.Supp.2d 1185. S.D. Fla.

Amstrup, S., McDonald, T. and Manly, B. (2005) *Handbook of Capture-Recapture Analysis*. Princeton University Press. URLhttp://books.google.com/books?id=aK7V1qyDjygC.

Asher, J., Banks, D. L. and Scheuren, F. (2008) *Statistical methods for human rights / Jana Asher, David Banks, Fritz J. Scheuren, editors*. Springer New York.

Baker, C. L., Specter, S. and Kline, T. R. (2016) How not to manage a common benefit fund: Allocating attorneys' fees in vioxx litigation. *Drexel Law Review*, **9**, 1.

Bar-Gill, O., Ben-Shahar, O. and Marotta-Wurgler, F. (2017) Searching for the common law: The quantitative approach of the restatement of consumer contracts. *University of Chicago Law Review*, **84**, 7–35.

Bartolucci, F. and Forcina, A. (2001) Analysis of capture-recapture data with a Rasch-type model allowing for conditional dependence and multidimensionality. *Biometrics*, **57**, 714–719. URLhttp://dx.doi.org/10.1111/j.0006-341X.2001.00714.x.

Baude, W., Chilton, A. S. and Malani, A. (2017) Making doctrinal work more rigorous: Lessons from systematic reviews. *University of Chicago Law Review*, **84**, 37–58.

Bishop, Y., Fienberg, S. and Holland, P. (1975) *Discrete Multivariate Analysis: Theory and Practice*. MIT Press. URLhttp://books.google.com/books?id=nPkjIEVY-CsC.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003) Latent dirichlet allocation. *J. Mach. Learn. Res.*, **3**, 993–1022. URLhttp://dl.acm.org/citation.cfm?id=944919.944937.

Bommarito II, M. J., Katz, D. and Zelner, J. (2009) Law as a seamless web?: comparison of various network representations of the united states supreme court corpus (1791-2005). In *Proceedings of the 12th international conference on artificial intelligence and law*, 234–235. ACM.

Brams, S. and Taylor, A. (1996) *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press. URLhttps://books.google.com/books?id=cLUA-sRhJ5QC.

Brooks v. State (1999) 748 So.2d 736. Miss.

Brown, L. D., Eaton, M. L., Freedman, D. A., Klein, S. P., Olshen, R. A., Wachter, K. W., Wells, M. T. and Ylvisaker, D. (1999) Statistical controversies in census 2000. *Jurimetrics*, **39**, 347–375. URLhttp://www.jstor.org/stable/29762618.

Chojnacki, D. E., Cicchini, M. D. and White, L. T. (2008) Empirical basis for the admission of expert testimony on false confessions, an. *Ariz. St. LJ*, **40**, 1.

Clark, T. S. and Lauderdale, B. (2010) Locating supreme court opinions in doctrine space. *American Journal of Political Science*, **54**, 871–890. URLhttp://dx.doi.org/10.1111/j.1540-5907.2010.00470.x.

de Clippel, G., Moulin, H. and Tideman, T. (2008) Impartial division of a dollar. *Journal of Economic Theory*, **139**, 176–191.

Condliff, M. K., Lewis, D. D., Madigan, D. and Posse, C. (1999) Bayesian Mixed-Effects Models for Recommender Systems. *ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*.

Crawford v. Washington (2004) 541 U.S. 36.

Darroch, J. N., Fienberg, S. E., Glonek, G. F. V. and Junker, B. W. (1993) A three-sample multiple-recapture approach to census population estimation with heterogeneous catchability. *Journal of the American Statistical Association*, **88**, pp. 1137–1148. URLhttp://www.jstor.org/stable/2290811.

Delgado, R. and Stefancic, J. (1989) Why do we tell the same stories?: Law reform, critical librarianship, and the triple helix dilemma. *Stanford Law Review*, **42**, 207.

Dryden, I. and Mardia, K. (1998) *Statistical Shape Analysis*. Wiley.

Dunn, E. and Ruiz, M. (2015) Can case citations tell us what a legal opinion is about? if so, why not use them to sort legal information?

Duval, S. and Tweedie, R. (2000) Trim and fill: A simple funnel-plot–based method of testing and adjusting for publication bias in meta-analysis. *Biometrics*, **56**, 455–463. URLhttp://dx.doi.org/10.1111/j.0006-341X.2000.00455.x.

Easterbrook, P., Gopalan, R., Berlin, J. and Matthews, D. (1991) Publication bias in clinical research. *The Lancet*, **337**, 867 – 872. URLhttp://www.sciencedirect.com/science/article/pii/014067369190201Y. Originally published as Volume 1, Issue 8746.

Eisenberg, T. and Miller, G. P. (2010) Attorneys' fees and expenses in class action settlements: 1993-2008. *Journal of Empirical Legal Studies*, **7**, 248–281.

Fabricant, M. C. and Carrington, T. (2016) The shifted paradigm: Forensic science's overdue evolution from magic to law. *Virgnia Journal of Criminal Law*, **4**, 1.

Fienberg, S. E., Johnson, M. S. and Junker, B. W. (1999) Classical multilevel and Bayesian approaches to population size estimation using multiple lists. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **162**, 383–405. URL`http://dx.doi.org/10.1111/1467-985X.00143`.

Fienberg, S. E. and Manrique-Vallier, D. (2009) Integrated methodology for multiple systems estimation and record linkage using a missing data formulation. *AStA Advances in Statistical Analysis*, **93**, 49–60. URL`http://dx.doi.org/10.1007/s10182-008-0084-z`.

Fitzpatrick, B. T. (2010a) Do class action lawyers make too little. *University of Pennsylvania Law Review*, **158**, 2043–2083.

— (2010b) An empirical study of class action settlements and their fee awards. *Journal of Empirical Legal Studies*, **7**, 811–846.

— (2010c) An empirical study of class action settlements and their fee awards. *Journal of Empirical Legal Studies*, **7**, 811–846.

Fowler, J., Johnson, T., Spriggs, J., Jeon, S. and Wahlbeck, P. (2007) Network analysis and the law: Measuring the legal importance of supreme court precedents. *Political Analysis*, **15**, 324–346.

Fowler, J. H. and Jeon, S. (2008) The authority of supreme court precedent. *Social networks*, **30**, 16–30.

Gartrell, M., Paquet, U. and Herbrich, R. (2012) A bayesian treatment of social links in recommender systems. *Tech. Rep. CU-CS-1092-12*, University of Colorado Department of Computer Science.

Gelman, A. (2006) Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian Anal.*, **1**, 515–534. URL`https://doi.org/10.1214/06-BA117A`.

Givens, G. H., Smith, D. D. and Tweedie, R. L. (1997) Publication bias in meta-analysis: a bayesian data-augmentation approach to account for issues exemplified in the passive smoking debate. *Statist. Sci.*, **12**, 221–250. URL`http://dx.doi.org/10.1214/ss/1030037958`.

Harvey, A. and Woodruff, M. J. (2011) Confirmation bias in the united states supreme court judicial database. *Journal of Law, Economics, and Organization*, **29**, 414–460.

Hoff, P. D., Raftery, A. E. and Handcock, M. S. (2001) Latent space approaches to social network analysis. *Journal of the American Statistical Association*, **97**, 1090–1098.

— (2002) Latent space approaches to social network analysis. *Journal of the American Statistical Association*, **97**, 1090–1098. URL`http://dx.doi.org/10.1198/016214502388618906`.

Holmes, O. W. (1897) The path of the law. *Harvard Law Review*, **10**, 457.

Hubbard, W. H. J. (2013) Testing for change in procedural standards, with application to bell atlantic v. twombly. *The Journal of Legal Studies*, **42**, 35–68. URL`https://doi.org/10.1086/668506`.

— (2017) The effects of Twombly and Iqbal. *Journal of Empirical Legal Studies*, **14**, 474–526. URL`http://dx.doi.org/10.1111/jels.12153`.

In re Initial Public Offering Securities Litigation (2011) 2011 WL 2732563. S.D.N.Y.

In re TFT-LCD (Flat Panel) Antitrust Litigation (2013) 2013 WL 1365900. N.D. Cal.

In re Vitamins Antitrust Litigation (2005) 398 F.Supp.2d 209. D.D.C.

Jasra, A., Holmes, C. C. and Stephens, D. A. (2005) Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*, **20**, 50–67. URL`http://dx.doi.org/10.1214/088342305000000016`.

Kassin, S. M. and Kiechel, K. L. (1996) The social psychology of false confessions: Compliance, internalization, and confabulation. *Psychological Science*, **7**, 125–128.

Kaye, D. H., Bernstein, D. E. and Mnookin, J. L. (2016) *The New Wigmore on Evidence: Expert Evidence*. Aspen. URL`http://books.google.com/books?id=aK7V1qyDjygC`.

King, R., Brooks, S. P., Mazzetta, C., Freeman, S. N. and Morgan, B. J. T. (2008) Identifying and diagnosing population declines: a bayesian assessment of lapwings in the uk. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **57**, 609–632. URL`http://dx.doi.org/10.1111/j.1467-9876.2008.00633.x`.

Laska, E. M. (2002) The use of capture-recapture methods in public health. *Bulletin of the World Health Organization*, **80**, 845–845. URL`http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2567685/`.

Ligons, M. J. (2000) Polygraph evidence: Where are we now? *Missouri Law Review*, **65**, 209–227.

Liu, J., Wu, C. and Liu, W. (2013) Bayesian probabilistic matrix factorization with social relations and item contents for recommendation. *Decision Support Systems*, **55**, 838–850.

Lizotte, B. N. (2007) Publish or perish: The electronic availability of summary judgments by eight district courts. *Wisconsin Law Review*, **2007**, 107–149.

Lum, K., Price, M. E. and Banks, D. (2013) Applications of multiple systems estimation in human rights research. *The American Statistician*, **67**, 191–200. URL`http://dx.doi.org/10.1080/00031305.2013.821093`.

Lunbery v. Hornbreak (2010) 605 F.3d 754. 9th Cir.

Lupu, Y. and Fowler, J. H. (2013) Strategic citations to precedent on the us supreme court. *Journal of Legal Studies*, **21**, 151–186.

Ma, H., Zhou, T. C., Lyu, M. R. and King, I. (2011) Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst.*, **29**, 9:1–9:23. URL`http://doi.acm.org/10.1145/1961209.1961212`.

Madigan, D., York, J. and Allard, D. (1995) Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, 215–232.

Madigan, D. and York, J. C. (1997) Bayesian methods for estimation of the size of a closed population. *Biometrika*, **84**, 19–31. URL`http://biomet.oxfordjournals.org/content/84/1/19.abstract`.

Martin, A. D. and Quinn, K. M. (2002) Dynamic ideal point estimation via markov chain monte carlo for the us supreme court, 1953–1999. *Political Analysis*, **10**, 134–153.

Merlo, P., Henderson, J., Schneider, G. and Wehrli, E. (2013) Learning document similarity using natural language processing. *Linguistik Online*, **17**. URL`https://bop.unibe.ch/linguistik-online/article/view/788`.

Merritt, D. J. and Brudney, J. J. (2001) Stalking secret law: What predicts publication in the united states courts of appeals. *Vanderbilt Law Review*, **54**, 71–121.

Moulin, H. (1991) *Axioms of Cooperative Decision Making*. Econometric Society Monographs. Cambridge University Press. URL`https://books.google.com/books?id=mK6nEvHnqQIC`.

Nolan, J. M., Schultz, P. W., Cialdini, R. B., Goldstein, N. J. and Griskevicius, V. (2008) Normative social influence is underdetected. *Personality and Social Psychology Bulletin*, **34**, 913–923. URL`http://psp.sagepub.com/content/34/7/913.abstract`.

Ofshe, Richard, J. and Leo, R. A. (1997) The decision to confess falsely: Rational choice and irrational action. *Denver University Law Review*, **74**, 979–1122.

Pelle, E., Hessen, D. J. and van der Heijden, P. G. M. (2016) A log-linear multidimensional Rasch model for capture–recapture. *Statistics in Medicine*, **35**, 622–634. URL`http://dx.doi.org/10.1002/sim.6741`. Sim.6741.

Rabina, D. L. and Sula, C. (2015) Visual first amendment: Using empirical legal methods and visualization techniques to enhance understanding of supreme court rulings. *iConference 2015 Proceedings*.

Regal, R. R. and Hook, E. B. (1984) Goodness-of-fit based confidence intervals for estimates of the size of a closed population. *Stat Med*, **3**, 287–291.

Risinger, D. M. (2007) Goodbye to all that, or a fool's errand, by one of the fools: How i stopped worrying about court responses to handwriting identification (and "forensic science" in general) and learned to love misinterpretations of kumho. *Tulsa Law Review*, **43**, 447–475.

Roberts, C. and Stanley, T. (2006) *Meta-Regression Analysis: Issues of Publication Bias in Economics*. Surveys of Recent Research in Economics. Wiley. URL`http://books.google.com/books?id=_oMTN0U9usAC`.

Robertson, J. and Webb, W. (1998) *Cake-Cutting Algorithms: Be Fair if You Can*. Ak Peters Series. Taylor & Francis. URL`https://books.google.com/books?id=XBCOmGFmpBMC`.

Rosenthal, R. (1979) The file drawer problem and tolerance for null results. *Psychological Bulletin*, **86**, 638–641.

Rubenstein, W. B. (2009) On what a common benefit is, is not, and should be. *Class Action Attorney Fee Digest*, **3**, 87–94.

Ruger, T. W., Kim, P. T., Martin, A. D. and Quinn, K. M. (2004) The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. *Columbia Law Review*, 1150–1210.

Seybolt, T., Aronson, J. and Fischhoff, B. (2013) *Counting Civilian Casualties: An Introduction to Recording and Estimating Nonmilitary Deaths in Conflict*. Studies in Strategic Peacebuilding. OUP USA. URL`https://books.google.com/books?id=9WJKkgEACAAJ`.

Sherry, S. (2004) What's law got to do with it? *Perspectives on Politics*, **2**, 769–775.

Siegelman, P. and Donohue, J. J. (1990) Studying the iceberg from its tip: A comparison of published and unpublished employment discrimination cases. *Law and Society Review*, 1133–1170.

Silver, C. and Miller, G. P. (2010) The quasi-class action method of managing multidistrict litigations: Problems and a proposal. *Vanderbilt Law Review*, **63**, 107.

Simonsohn, U., Nelson, L. D. and Simmons, J. P. (2014) p-Curve and effect size. *Perspectives on Psychological Science*, **9**, 666–681. URL`https://doi.org/10.1177/1745691614553988`.

Spaeth, H., Epstein, L., Ruger, T., Whittington, K., Segal, J. and Martin, A. D. (2013) Supreme court database.

Stan (2015) Stan: A c++ library for probability and sampling, version 2.7.0. URL`http://mc-stan.org/`.

Stanghellini, E. and van der Heijden, P. G. M. (2004) A multiple-record systems estimation method that takes observed and unobserved heterogeneity into account. *Biometrics*, **60**, 510–516. URLhttp://dx.doi.org/10.1111/j.0006-341X.2004.00197.x.

State v. Buechler (1998) 572 N.W.2d 65. Neb.

Stith, K. (1990) The risk of legal error in criminal cases: Some consequences of the asymmetry in the right to appeal. *University of Chicago Law Review*, **57**, 1–61.

Tideman, T. N. and Plassmann, F. (2008) Paying the partners. *Public Choice*, **136**, 19–37. URLhttps://doi.org/10.1007/s11127-008-9276-z.

United States v. Brito (2005) 427 F.3d 53. 1st Cir.

United States v. Hadley (2005) 431 F.3d 484. 6th Cir.

Vent v. State (2003) 67 P.3d 661. Alaska Ct. App.

Victor v. Argent Classic Convertible Arbitrage Fund L.P. (2010) 623 F.3d 82. 2d Cir.

Wininger, P. J. and Cecil, J. S. (2015) Finding motions: Comparison of samples of 12(b)(6) orders from westlaw, dockets, and cm/ecf codes.

Appendix

Code for Chapter 1 (Publication Bias)

R Code

Simulation

```r
# Script File
# Simulated Publication Bias
# Edward K. Cheng

# Initialization
rm(list = ls())    # clear all
name <- "papersim1a"

# HPC Specific Commands
.libPaths("/vega/stats/users/ekc2109/rpackages/")   # Use for cluster
library(rstan)
set_cppo("fast")

# Local Specific Commands
# setwd("/Users/ekcheng/Documents/PubBias/HPC/Sim7a")
library(rstan)
set_cppo('debug')   # to make debugging easier
library(stats)
library(foreign)
library(MASS)
library(reshape)

# FUNCTION
# Takes a vector of binary values
# Generates Bernoulli variable X conditional on data vector dta)
# p_0 = P(X=1 | dta=0)
# p_1 = P(X=1 | dta=1)

genCondBern <- function (dta, p_0, p_1) {
  N_dta <- length(dta)
  N_0 <- sum(dta==0)
  N_1 <- sum(dta==1)

  newset <- rep(NA, N_dta)

  newset_0 <- rbinom(N_0, 1, p_0)
  newset[dta==0] <- newset_0

  newset_1 <- rbinom(N_1, 1, p_1)
  newset[dta==1] <- newset_1

  return(newset)
  }

# 2.  SIMULATION PARAMETERS
set.seed(2813)
N_case <- 250
p_admit <- 0.33        # True probability of admission

p_obs_admit_a <- 0.02
p_obs_excl_a <- 0.1

p_obs_admit_t <- 0.001
p_obs_excl_t <- 0.005

# 2.1 Construct outcomes
rawset <- data.frame(case=paste("Case", c(1:N_case), sep=""))
```

```r
rawset$admis <- rbinom(N_case, 1, p_admit)

# 2.2 Observation Data (Independent)
Published1 <- genCondBern(rawset$admis, 2*p_obs_excl_t, 2*p_obs_admit_t)
rawset$ListA1 <- genCondBern(rawset$admis, p_obs_excl_t, p_obs_admit_t)
rawset$ListB1 <- genCondBern(rawset$admis, 2 * p_obs_excl_t, 2 * p_obs_admit_t)
rawset$ListC1<- genCondBern(rawset$admis, 4 * p_obs_excl_t, 4 * p_obs_admit_t)
rawset$SimLex1 <- genCondBern(rawset$admis, 2 * p_obs_excl_t, 2 * p_obs_admit_t) | Published1
rawset$SimWest1 <- genCondBern(rawset$admis, 4 * p_obs_excl_t, 4 * p_obs_admit_t) | Published1
rawset$Published1 <- Published1

Published2 <- genCondBern(rawset$admis, 2*p_obs_excl_a, 2*p_obs_admit_a)
rawset$ListA2 <- genCondBern(rawset$admis, p_obs_excl_a, p_obs_admit_a)
rawset$ListB2 <- genCondBern(rawset$admis, 2 * p_obs_excl_a, 2 * p_obs_admit_a)
rawset$ListC2<- genCondBern(rawset$admis, 4 * p_obs_excl_a, 4 * p_obs_admit_a)
rawset$SimLex2 <- genCondBern(rawset$admis, 2 * p_obs_excl_a, 2 * p_obs_admit_a) | Published2
rawset$SimWest2 <- genCondBern(rawset$admis, 4 * p_obs_excl_a, 4 * p_obs_admit_a) | Published2
rawset$Published2 <- Published2

rawset$agg <- rowSums(rawset[,-c(1:2)]) > 0

culledset <- rawset[rawset$agg == TRUE, ]
culledset$agg <- NULL

nrow(culledset)
mean(rawset$admis)      # Actual admissibility rate
mean(culledset$admis)     # Observed admissibility rate

# 2.3 Create set for MCMC
combinedset <- culledset

meltset <- melt(combinedset, id=c("case", "admis"), variable_name="listtype")
colnames(meltset)[4] <- "obs"

meltset$list <- sub("[0-9]", "", as.character(meltset$listtype))          # Create List and Type columns in dataset
meltset$type <- sub("[A-Za-z]*", "", perl=TRUE, as.character(meltset$listtype))

fcset <- meltset
fcset$case <- factor(fcset$case)
fcset$list <- factor(fcset$list, levels=c("ListA", "ListB", "ListC", "SimLex", "SimWest", "Published"))
fcset$listtype <- factor(fcset$listtype)
fcset$type <- as.integer(fcset$type)      # Keep as number for easy indexing

levels(fcset$list)

# Create Z array
# It is the inverse of the Published list vector as required by the model
Z <- rbind(
    1-combinedset$"Published1",
    1-combinedset$"Published2")

# 2.4 Create "dataset"
# Case, type -- then 0/1 for the lists
dataset <- fcset
dataset$listtype <- NULL

dataset <- reshape(dataset, timevar="list", idvar=c("case","admis", "type"), direction="wide")
colnames(dataset) <- sub("obs.", "", perl=TRUE, as.character(colnames(dataset)))

dataset <- unique(dataset)
dataset_obs <- rowSums(dataset[,4:9])
```

```r
dataset <- dataset[dataset_obs > 0, ]

dataset[with(dataset, order(dataset$case, dataset$type)), ]

# 3. MULTIPLE SYSTEMS ESTIMATION
fc_dat <- list (
    K = length(levels(fcset$case)),
    L = length(levels(fcset$list)),
    T = max(fcset$type),

    N = nrow(fcset),
    P = as.integer(fcset$obs),
    casename = as.integer(fcset$case),
    type = as.integer(fcset$type),
    list = as.integer(fcset$list),
    listtype = as.integer(fcset$listtype),
    admis = as.integer(fcset$admis),

    Z = Z

    )
ptm <- proc.time()
mse_fit <- stan(file =  paste(name, ".stan", sep=""), data = fc_dat,
             iter = 5000, warmup=1000, chains = 1)
proc.time() - ptm

# 4. POST-PROCESSING MATERIALS
dataset$type <- factor(dataset$type)      # Factor them so that tables below will have the zero entries
dataset$case <- factor(dataset$case)

p_obs <- mean(combinedset$admis)
tau_wt <- table(dataset$type)/nrow(dataset)
gamma_wt <- table(dataset$case)/nrow(dataset)
tautheta_wt <- table(dataset$type, dataset$case) / nrow(dataset)

dataset_a <- dataset[dataset$admis == 1, ]
dataset_abar <- dataset[dataset$admis == 0, ]

tautheta_wt_a <- table(dataset_a$type, dataset_a$case) / nrow(dataset_a)
tautheta_wt_abar <- table(dataset_abar$type, dataset_abar$case) / nrow(dataset_abar)

# 4.1 GET TYPE COMBINATION FREQUENCIES
library(sets)
library(plyr)

# Get Count Matrix
typedata <- dataset[,c(1,3)]
typedata$value <- 1
typegrid <- cast(typedata, case ~ type, fill=0)
typegrid$numobs <- rowSums(typegrid[,-1])              # Max number of shared = 3

typecounts <- count(typegrid[,-1])

T <- 2

# Singles
singles <- data.frame(1:T)
singles$freq <- NULL
for (i in 1:nrow(singles)) {
  counts <- typecounts[(typecounts[,singles[i,1]] == 1), ]
  singles$freq[i] <- sum(counts$freq)
  }
singles$freq <- singles$freq / nrow(typegrid)

  #typefreq1b <- colSums(typegrid[,-1])/nrow(typegrid)        # Matches

# Pairs
```

```
pairs <- set_combn(set(1,2), 2)
pairs <- as.data.frame(t(matrix(unlist(pairs), nrow=2)))    # Coerces the pairs into
      data frame
pairs$freq <- NULL

for (i in 1:nrow(pairs)) {
  counts <- typecounts[(typecounts[,pairs[i,1]] == 1) & (typecounts[,pairs[i,2]] ==
    1), ]
# print(c(pairs[i,1], pairs[i,2]))
# print(counts)
  pairs$freq[i] <- sum(counts$freq)
  }
pairs$freq <- pairs$freq / nrow(typegrid)


# True admissibility rate
p_actual <- mean(rawset$admis)


# 5. SAVE AND REPORT

mse_fit
save(mse_fit, p_actual, p_obs, tau_wt, gamma_wt, tautheta_wt, tautheta_wt_a,
    tautheta_wt_abar, singles, pairs, file=paste(name, ".Rdata", sep=""))
```

Simulation Analysis

```
# Script File
# Simulated Publication Bias Dataset
# Edward K. Cheng

# Initialization
rm(list = ls())    # clear all

name <-"papersim1"

setwd(paste("/Users/ekcheng/Documents/PubBias/HPC/", name, sep=""))

library(rstan)
set_cppo('debug')   # to make debugging easier

library(stats)
library(foreign)
library(MASS)
library(reshape)
library(coda)
library(boot)
library(polynom)
library(rstan)

### FUNCTIONS
# Conversion function (thanks to Ben Goodrich)
stan2coda <- function(fit) {
      mcmc.list(lapply(1:ncol(fit), function(x) mcmc(as.array(fit)[,x,])))
}

loadbeta <- function(name) {
  load(paste(name, ".Rdata", sep=""))
  extract(mse_fit, "beta")$beta
}

plotdens <- function (dataset, color, width=1) {
  dens.pts <- density(dataset, kernel="gaussian")
  lines(dens.pts, col=color, lwd=width)
  }

# RECOVERY OF ADMISSIBILITY PROBABILITY
```

```
# EXPLORE ROOTS

load(paste(name, "a.Rdata", sep=""))
  # Loads mse_fit, taugamma_wt[T,K], p_obs, tau_wt[T]

  draws <- extract(mse_fit)
  N <- length(draws$beta)

  p_corr <- rep(NA, N)
  lik_num <- rep(NA, N)
  lik_denom <- rep(NA, N)

  L <- 5

  results_denom <- matrix(nrow=N, ncol=(6 + 1  + 5))   # ncol = # of lists + zero + # of roots
  results_num <- matrix(nrow=N, ncol=(6 + 1  + 5))

  for (i in 1:N) {
    beta <- draws$beta[i]
    gamma <- draws$gamma_list[i,]
    theta <- draws$theta_case[i,]
    rho <- draws$rho[i,]
    tau <- draws$tau_type[i,]        # ONLY WORKS FOR DIRECT APPEAL CASES
    P <- length(gamma)
    L <- P-1
    K <- length(theta)
    T <- length(tau)


# INCORPORATES THETA
# R_A (Denominator)

    R_A <- rep(NA, P)

    for (l in 1:P) {
      R_A_t <- rep(NA, T)

      # Integrate over k
      for (t in 1:T) {
        R_A_t[t] <- sum(inv.logit(beta + gamma[l] + theta + tau[t] + rho[l])) / K
        }

      R_A_singles <- singles
        R_A_singles$O_l <- R_A_t[R_A_singles[,1]]
        R_A_singles$prod <- R_A_singles$O_l * R_A_singles$freq

      R_A_pairs <- pairs
        R_A_pairs$O_l <- R_A_t[R_A_pairs[,1]] * R_A_t[R_A_pairs[,2]]
        R_A_pairs$prod <- R_A_pairs$O_l * R_A_pairs$freq

      R_A[l] <- sum(R_A_singles$prod) - sum(R_A_pairs$prod)

      }

    right <- polynomial(coef = 1)

      for (l in 1:L) {
        regterm <- polynomial(coef = c(1, -(R_A[P] + R_A[l] - R_A[P]*R_A[l])) )
        right <- right * regterm
        }

    left <- polynomial(coef = c(1, -1)) * ( (polynomial(coef = c(1, -R_A[P])))^(L-1) )

    p <- left - right

```

```r
    zeroes <- solve(p)

    zero <- NA
    j <- 1
    while (is.na(zero) & (j <= length(zeroes)) ) {
      z <- zeroes[j]
      if ( (Im(z) == 0) & (Re(z) > 0) & (Re(z) < 1) ) { zero <- z }
      j <- j+1
      }

    results_denom[i,] <- c(R_A, zero, zeroes)

    lik_denom[i] <- Re(zero)

# R_Abar (Numerator)

    R_Abar <- rep(NA, P)

    for (l in 1:P) {
      R_Abar_t <- rep(NA, T)

      # Integrate over k
      for (t in 1:T) {
        R_Abar_t[t] <- sum(inv.logit(gamma[l] + theta + tau[t] + rho[l])) / K
        }

      R_Abar_singles <- singles
        R_Abar_singles$O_l <- R_Abar_t[R_Abar_singles[,1]]
        R_Abar_singles$prod <- R_Abar_singles$O_l * R_Abar_singles$freq

      R_Abar_pairs <- pairs
        R_Abar_pairs$O_l <- R_Abar_t[R_Abar_pairs[,1]] * R_Abar_t[R_Abar_pairs[,2]]
        R_Abar_pairs$prod <- R_Abar_pairs$O_l * R_Abar_pairs$freq

      R_Abar[l] <- sum(R_Abar_singles$prod) - sum(R_Abar_pairs$prod)

      }

    right <- polynomial(coef = 1)

      for (l in 1:L) {
        regterm <- polynomial(coef = c(1, -(R_Abar[P] + R_Abar[l] - R_Abar[P]*R_Abar[l])) )
        right <- right * regterm
        }

    left <- polynomial(coef = c(1, -1)) * ( (polynomial(coef = c(1, -R_Abar[P])))^(L-1) )

    p <- left - right

    zeroes <- solve(p)

    zero <- NA
    j <- 1
    while (is.na(zero) & (j <= length(zeroes)) ) {
      z <- zeroes[j]
      if ( (Im(z) == 0) & (Re(z) > 0) & (Re(z) < 1) ) { zero <- z }
      j <- j+1
      }

    results_num[i,] <- c(R_Abar, zero, zeroes)

    lik_num[i] <- Re(zero)

# Correction
    odds_corr <- (lik_num[i]/ lik_denom[i]) * (p_obs / (1-p_obs))
    p_corr[i] <- odds_corr/ (1 + odds_corr)
```

```
# Counter
    if (i %% 1000 == 0) {print(i)}
  }



# DIAGNOSTICS

results <- data.frame(results_num)    # Choose which set to examine

colnames(results) <- c("R_1", "R_2", "R_3", "R_4", "R_5", "R_P", "zero", "z1", "z2",
     "z3", "z4", "z5")
for (i in 1:5) {
  results[,i] <- Re(results[,i])
  }
results$probsum <- rowSums(results[,1:5])
# Re(colMeans(results[,1:7]))


data.frame(results$zero, results$probsum)   # Shows that loss of root is due to lack
     of coverage of the probability space (O*)


# CORRECTION
  p_corr <- p_corr[!is.na(p_corr)]
  lik_num <- lik_num[!is.na(lik_num)]
  lik_denom <- lik_denom[!is.na(lik_denom)]

  data.frame(p_obs=p_obs, mean=mean(p_corr), med=median(p_corr), lik_num=mean(lik_
    num),
       lik_denom=mean(lik_denom))

# EXPLORATION

colSums(dataset[,4:10]) / nrow(dataset)

## PLOTS
png(paste("Analysis beta ",name, ".png", sep=""), 500,500)
plot(1, pch=19, cex=0.3, xlim=c(-2, 2), ylim=c(0,1.5),
     xlab="beta", ylab="Density", main="Offdata13")
plotdens(draws$beta, "red")
dev.off()

png(paste("Analysis pcorr ",name, ".png", sep=""), 500,500)
plot(1, pch=19, cex=0.3, xlim=c(0, 1), ylim=c(0,3),
     xlab="pcorr", ylab="Density", main="Offdata13")
plotdens(p_corr, "red")
dev.off()
```

False Confession Expert Testimony

```
# Script File
# Simulated Publication Bias Dataset
# Edward K. Cheng

# Initialization
rm(list = ls())    # clear all

name <- "offdata13"

# HPC Specific Commands
.libPaths("/vega/stats/users/ekc2109/rpackages/")   # Use for cluster
library(rstan)
set_cppo("fast")
library(stats)
```

```r
library(foreign)
library(MASS)
library(reshape)

set.seed(2813)

# FUNCTION
genCondBern <- function (dta, p_0, p_1) {
  N_dta <- length(dta)
  N_0 <- sum(dta==0)
  N_1 <- sum(dta==1)

  newset <- rep(NA, N_dta)

  newset_0 <- rbinom(N_0, 1, p_0)
  newset[dta==0] <- newset_0

  newset_1 <- rbinom(N_1, 1, p_1)
  newset[dta==1] <- newset_1

  return(newset)
  }

# 1. IMPORT DATASET

# 1.1 Importation
dataraw <- read.csv("fcdatasetv3.csv")

dataset <- dataraw[,c(2, 19, 25, 5:6, 4, 7:11)]       # Reorder and get relevant columns
colnames(dataset)[1:3] <- c("case", "type", "admis")

colnames(dataset)[7] <- "Google"
colnames(dataset)[11] <- "DaubertTracker"

dataset$case <- paste("Case",dataset$case, sep="")

# 2. PREP DATASET FOR USE

# Ensure unique
uniqset <- unique(data.frame(dataset$case, admis=dataset$admis))
nrow(uniqset)           # Number of unique cases: 110 (lost nothing)
nrow(dataset)           # Number of observed cases (trial and app counted individually): 136

reducedset <- dataset[dataset$type==1, ]

# Observation Rate
p_obs <- mean(reducedset$admis)

# 2.2 Reorder
# Remove Findlaw, Remove type
dataset <- dataset[c("case", "admis", "type", "DaubertTracker", "Fastcase", "BNA", "Google", "Lexis", "Westlaw", "Published")]

# 2.3 Infer Gaps

# 2.3.1 Break-up dataset into different types
# Convert the numbering from 0 through 4 to 1 through 5 because then it will enable easy index referencing

dataset$type <- dataset$type+1

data1 <- dataset[dataset$type==1,]
  colnames(data1)[4:10] <- paste(colnames(dataset)[4:10], "1", sep="")
  data1 <- data1[,-3]     # Removes type, which we now have coded into column
data2 <- dataset[dataset$type==2,]
  colnames(data2)[4:10] <- paste(colnames(dataset)[4:10], "2", sep="")
```

```r
  data2 <- data2[,-3]
data3 <- dataset[dataset$type==3,]
  colnames(data3)[4:10] <- paste(colnames(dataset)[4:10], "3", sep="")
  data3 <- data3[,-3]
data4 <- dataset[dataset$type==4,]
  colnames(data4)[4:10] <- paste(colnames(dataset)[4:10], "4", sep="")
  data4 <- data4[,-3]
data5 <- dataset[dataset$type==5,]
  colnames(data5)[4:10] <- paste(colnames(dataset)[4:10], "5", sep="")
  data5 <- data5[,-3]

# 2.3.2 Merge full datasets based on number of hierarchical levels to search
merge12 <- merge(data1, data2, by=c("case", "admis"), all=TRUE)
merge12[is.na(merge12)] <- 0        # Replace NA with 0 (because not seen)
merge12$case <- factor(merge12$case)    # Reset the factor

merge123 <- merge(merge12, data3, by=c("case", "admis"), all=TRUE)
merge123[is.na(merge123)] <- 0
merge123$case <- factor(merge123$case)    # Reset the factor

mergeall <- merge(merge123, data4, by=c("case", "admis"), all=TRUE)
mergeall[is.na(mergeall)] <- 0
mergeall <- merge(mergeall, data5, by=c("case", "admis"), all=TRUE)
mergeall[is.na(mergeall)] <- 0
mergeall$case <- factor(mergeall$case)    # Reset the factor

# 2.4 Melt

combinedset <- mergeall     # Take all lists

meltset <- melt(combinedset, id=c("case", "admis"), variable_name="listtype")
colnames(meltset)[4] <- "obs"

meltset$list <- sub("[0-9]", "", as.character(meltset$listtype))          # Create List and Type columns in dataset
meltset$type <- sub("[A-Za-z]*", "", perl=TRUE, as.character(meltset$listtype))

fcset <- meltset
fcset$case <- factor(fcset$case)
fcset$list <- factor(fcset$list, levels=c("BNA", "DaubertTracker", "Fastcase", "Google", "Lexis", "Westlaw", "Published"))
fcset$listtype <- factor(fcset$listtype)
fcset$type <- as.integer(fcset$type)      # Keep as number for easy indexing

levels(fcset$list)

# 2.5 Create Z array
# It is the inverse of the Published list vector as required by the model
Z <- rbind(
    1-mergeall$"Published1",
    1-mergeall$"Published2",
    1-mergeall$"Published3",
    1-mergeall$"Published4",
    1-mergeall$"Published5")

# Check
# mergeall$case == levels(fcset$case)
# data.frame(mergeall$case, levels(fcset$case))

# 3. MULTIPLE SYSTEMS ESTIMATION
fc_dat <- list (
    K = length(levels(fcset$case)),
    L = length(levels(fcset$list)),
    T = max(fcset$type),

    N = nrow(fcset),
    P = as.integer(fcset$obs),
    casename = as.integer(fcset$case),
```

```
    type = as.integer(fcset$type),
    list = as.integer(fcset$list),
    listtype = as.integer(fcset$listtype),
    admis = as.integer(fcset$admis),

    Z = Z

    )
ptm <- proc.time()
mse_fit <- stan(file =  paste(name, ".stan", sep=""), data = fc_dat,
            iter = 5000, warmup=1000, chains = 1)
proc.time() - ptm


# 4. POST-PROCESSING MATERIALS

dataset$type <- factor(dataset$type)      # Factor them so that tables below will
    have the zero entries
dataset$case <- factor(dataset$case)

p_obs <- mean(combinedset$admis)
tau_wt <- table(dataset$type)/nrow(dataset)
gamma_wt <- table(dataset$case)/nrow(dataset)
tautheta_wt <- table(dataset$type, dataset$case) / nrow(dataset)


dataset_a <- dataset[dataset$admis == 1, ]
dataset_abar <- dataset[dataset$admis == 0, ]

tautheta_wt_a <- table(dataset_a$type, dataset_a$case) / nrow(dataset_a)
tautheta_wt_abar <- table(dataset_abar$type, dataset_abar$case) / nrow(dataset_abar)

# 4.1 GET TYPE COMBINATION FREQUENCIES
library(sets)
library(plyr)

# Get Count Matrix
typedata <- dataset[,c(1,3)]
typedata$value <- 1
typegrid <- cast(typedata, case ~ type, fill=0)
typegrid$numobs <- rowSums(typegrid[,-1])             # Max number of shared = 3

typecounts <- count(typegrid[,-1])

T <- 5

# Singles
singles <- data.frame(1:T)
singles$freq <- NULL
for (i in 1:nrow(singles)) {
  counts <- typecounts[(typecounts[,singles[i,1]] == 1), ]
  singles$freq[i] <- sum(counts$freq)
  }
singles$freq <- singles$freq / nrow(typegrid)

  #typefreq1b <- colSums(typegrid[,-1])/nrow(typegrid)         # Matches

# Pairs
pairs <- set_combn(set(1,2,3,4,5), 2)
pairs <- as.data.frame(t(matrix(unlist(pairs), nrow=2)))    # Coerces the pairs into
     data frame
pairs$freq <- NULL

for (i in 1:nrow(pairs)) {
  counts <- typecounts[(typecounts[,pairs[i,1]] == 1) & (typecounts[,pairs[i,2]] ==
    1), ]
# print(c(pairs[i,1], pairs[i,2]))
# print(counts)
```

```
  pairs$freq[i] <- sum(counts$freq)
  }
pairs$freq <- pairs$freq / nrow(typegrid)

# Triples
trips <- set_combn(set(1,2,3,4,5), 3)
trips <- as.data.frame(t(matrix(unlist(trips), nrow=3)))    # Coerces the pairs into data frame
trips$freq <- NULL

for (i in 1:nrow(pairs)) {
  counts <- typecounts[(typecounts[,trips[i,1]] == 1) & (typecounts[,trips[i,2]] == 1) & (typecounts[,trips[i,3]] == 1), ]
  trips$freq[i] <- sum(counts$freq)
  }
trips$freq <- trips$freq / nrow(typegrid)


# 5. SAVE AND REPORT

mse_fit
save(mse_fit, p_obs, tau_wt, gamma_wt, tautheta_wt, tautheta_wt_a, tautheta_wt_abar, singles, pairs, trips, file=paste(name, ".Rdata", sep=""))
```

Stan Code

Simulation

```
# STAN model file for Publication Bias
# Correlated model


data {
  int<lower=1> K;               // number of cases
  int<lower=1> L;               // number of lists
  int<lower=1> N;               // number of vote observations
  int<lower=1> T;               // number of types
  int<lower=0, upper=1> P[N];          // observed votes
  int<lower=1, upper=K> casename[N];       // case for observation i
  int<lower=1, upper=T> type[N];         // type of case for observation i
  int<lower=1, upper=L> list[N];         // list for observation i
  int<lower=0, upper=1> admis[N];      // lower court admissibility determination for observation i
  int<lower=0, upper=1> Z[T,K];        // Inverse vector of official pub list
                          // Row is Official Pub List for the type, Column is case
  }


parameters {

  real beta;               // admis dependent intercept
  real theta_case[K];            // latent publication-worthiness of case k
  real gamma_list[L];            // latent comprehensiveness of list l
  real tau_type[T];            // latent noteworthiness of type t
  real<lower=0> sigma_case;        // hyperparameter for cases
  real<lower=0> sigma_type;
  real r[L-1];               // interaction
                        // for now assume lists have single rho
  real<lower=0> sigma_r;
  }

transformed parameters {
  real rho[L];

```

```
  // Build rho vector
  // Based on discussion on stan-users group
  // Assumes official publication list is the last list

  for (i in 1:(L-1)) {
    rho[i] <- r[i];
    }

  rho[L] <- 0;

}

model {
// hyperpriors
  sigma_case ~ cauchy(0,5);       // Half-cauchy prior suggested by Gelman(5006)
  sigma_type ~ cauchy(0,5);

  sigma_r ~ cauchy(0,5);

// priors
  theta_case ~ normal(0, sigma_case);
  tau_type ~ normal(0, sigma_type);       // Types are not linked
  for (l in 1:(L-1)) {
    gamma_list[l] ~ normal(0, 100);
    }
  gamma_list[L] ~ normal(0, 100);      // Let published list have different attribute
  beta ~ normal(0,100);
  r ~ normal(0, sigma_r);

// regressions
  for (i in 1:N) {
    P[i] ~ bernoulli_logit(
      beta * admis[i] + theta_case[casename[i]] + tau_type[type[i]] + gamma_list[
    list[i]]
      + rho[list[i]] * Z[type[i], casename[i]]
      );
    }
  }
```

False Confession Expert Testimony

```
# STAN model file for Publication Bias

data {
  int<lower=1> K;              // number of cases
  int<lower=1> L;              // number of lists
  int<lower=1> N;              // number of vote observations
  int<lower=1> T;              // number of types
  int<lower=0, upper=1> P[N];          // observed votes
  int<lower=1, upper=K> casename[N];       // case for observation i
  int<lower=1, upper=T> type[N];         // type of case for observation i
  int<lower=1, upper=L> list[N];         // list for observation i
  int<lower=0, upper=1> admis[N];      // lower court admissibility determination for
     observation i
  int<lower=0, upper=1> Z[T,K];        // Inverse vector of official pub list
                         // Row is Official Pub List for the type, Column is case
  }

parameters {
  real beta;               // admis dependent intercept
  real theta_case[K];            // latent publication-worthiness of case k
  real gamma_list[L];            // latent comprehensiveness of list l
  real tau_type[T];            // latent noteworthiness of type t
  real<lower=0> sigma_case;        // hyperparameter for cases
  real<lower=0> sigma_type;
  real r[L-1];               // interaction
                       // for now assume lists have single rho
```

```
  real<lower=0> sigma_r;

  }

transformed parameters {
  real rho[L];

  // Build rho vector
  // Based on discussion on stan-users group
  // Assumes official publication list is the last list

  for (i in 1:(L-1)) {
    rho[i] <- r[i];
    }

  rho[L] <- 0;

}


model {
// hyperpriors
  sigma_case ~ cauchy(0,5);        // Half-cauchy prior suggested by Gelman(5006)
  sigma_type ~ cauchy(0,5);
  sigma_r ~ cauchy(0,5);

// priors
  theta_case ~ normal(0, sigma_case);
  tau_type ~ normal(0, sigma_type);        // Types are not linked
  for (l in 1:(L-1)) {
    gamma_list[l] ~ normal(0, 100);
    }
  gamma_list[L] ~ normal(0, 100);      // Let published list have different attribute
  beta ~ normal(0,100);
  r ~ normal(0, sigma_r);

// regressions
  for (i in 1:N) {
    P[i] ~ bernoulli_logit(
      beta * admis[i] + theta_case[casename[i]] + tau_type[type[i]] + gamma_list[
     list[i]]
      + rho[list[i]] * Z[type[i], casename[i]]
      );
    }
  }
```

Code for Chapter 2 (Attorneys' Fees)

R Code

Simulations 1 and 2

```r
# Script File
# Fee Division Simulation 1 & 2
# Edward K. Cheng

# Initialization
rm(list = ls())    # clear all

# Local Specific Commands
setwd("/Users/ekcheng/Documents/Dissertation/Chapter2/Code")
library(rstan)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
library(stats)
library(foreign)
library(MASS)
library(reshape)


# 1. FUNCTIONS
# Both alr and alr.inv are checked

alr <- function(x) {
  d <- dim(x)
  if (d[1] != d[2]) print("WARNING (alr): Data matrix not square")
  N <- d[1]
  n <- N-1
  x_N <- x[,N]

  r <- matrix(NA, N, n)         # Creating normalized r vector
  for (i in 1:N) {
    r[i,1:n] <- x[i,1:n] /x_N[i]
    }
  lr <- log(r)                  # Take log

  lr
  }

alr.inv <- function(lr) {
  d <- dim(lr)
  if (d[1] != d[2] + 1) print("WARNING (alr): Transformed matrix not N x n")
  N <- d[1]
  n <- d[2]

  r <- exp(lr)
  x <- matrix(NA, N, N)
  x[,N] <- 1 / (1 + rowSums(r))
  for (i in 1:N) {
    x[i,1:n] <- r[i,1:n] * x[i,N]
    }
  x
  }

alr.single <- function(x) {
  N <- length(x)
  n <- N-1
  r <- x[1:n] /x[N]
  lr <- log(r)                  # Take log
  lr
}
```

```
alr.inv.single <- function(lr) {
  n <- length(lr)
  N <- n+1

  r <- exp(lr)
  x <- rep(NA, N)
  x[N] <- 1 / (1 + sum(r))
  x[1:n] <- r * x[N]
  x
  }

Fmatrix <- function (d) {
  cbind(diag(d), rep(-1,d))
}

Hmatrix <- function (d) {
  diag(d) + matrix(1,nrow=d,ncol=d)
}

Zmatrix <- function (scorevector) {
  w <- !is.na(scorevector)      # vector of non-missing values
  D <- length(scorevector)      # dimension of full simplex
  I <- diag(D)
  I[(w==1),]
}

getC <- function (scorevector) {
  C <- sum(!is.na(scorevector))
}

## Q Matrix -- Aitchison p.119
## Test by feeding it a basic vector of scores and see if it gives you the reduction
    matrix
Qmatrix <- function(scorevector) {
  w <- !is.na(scorevector)      # vector of non-missing values
  D <- length(scorevector)      # dimension of full simplex
  C <- sum(w)          # dimension of subcomposition simplex
  Z <- Zmatrix(scorevector)

  Fmatrix(C-1) %*% Z %*% t(Fmatrix(D-1)) %*% solve(Hmatrix(D-1))      # Results in
    correct result on test
}

# 3 STAN EXECUTION FUNCTION
stan_solve <- function(S, stancode, iterations) {
# Create Q and Z matricies -- to handle missing data
  F <- nrow(S)
  N <- ncol(S)

  Qindex <- rep(NA, F)
  #Qlist <- NULL
  Q <- NULL

  Zindex <- rep(NA, F)
  Zlist <- NULL
  Z <- NULL

  C <- rep(NA, F)

  # Loop
  for (i in 1:F) {
    newQ <- Qmatrix(S[i,])         # Round very small values
    newZ <- Zmatrix(S[i,])

    Q <- rbind(Q, newQ)
    Z <- rbind(Z, newZ)
    C[i] <- getC(S[i,])

```

```
    #Qlist[[i]] <- newQ                          # Test code
    Zlist[[i]] <- newZ

    if (i==1) {                                  # Sets the i=1
      Qindex[i] <- 1
      Zindex[i] <- 1
    }
    if (i < F) {                            # Sets the next index
      Qindex[i+1] <- Qindex[i] + C[i] - 1
      Zindex[i+1] <- Zindex[i] + C[i]
    }
  }

  S_all <- S                  # Do this to ease transfer of data to STAN
  S_all[is.na(S)] <- -99        # Missing values are tagged with -99 so we can detect
    errors in STAN

  simul_dat <- list (
    N = N,
    F = F,
    S_all = S_all,
    rater = c(1:F),

    Q_all = Q,
    Qindex = Qindex,
    Qlength = nrow(Q),

    Z_all = Z,
    Zindex = Zindex,
    Zlength = nrow(Z),

    C_all = C
  )

  ptm <- proc.time()
  stan_fit <- stan(file = stancode, data = simul_dat,
                   iter = iterations, chains = 4)
  proc.time() - ptm

  stan_fit
}

#4. EXTRACTION FUNCTION
postmode <- function (draws) {
    N <- ncol(draws)
    estmode <- rep(NA, N)
    for (i in 1:N) {
        dens <- density(draws[,i])
        estmode[i] <- dens$x[dens$y == max(dens$y)]
    }
    estmode
}

# 4. SIMULATION RUNS

# 4.1 Simulation 1
obs_data <- c( NA,    .5,   .4,   .1,
                .7,  NA,   .3,   NA,
               .85,  NA,   NA,   .15,
               NA,   0.5,   0.5,    NA)
obs <- t(matrix(obs_data, 4, 4))

# 4.1.1 Sim1 Linear
scores <- melt(t(obs))
colnames(scores) <- c("rated", "rater", "S")
scores <- scores[!is.na(scores$S),]
scores <- scores[,c(2,1,3)]
X <- t(matrix(as.integer(!is.na(obs_data)), 4, 4))          # What observations
```

```
     exist
D <- 4
simul_dat <- list (
    N = length(scores$S),
    S = scores$S,
    rater = scores$rater,
    rated = scores$rated,
    D = D,
    X = X,
    ones <- rep(1,D)
)

set.seed(888)
sim1_linear <- stan(file =  "linearmodel.stan", data = simul_dat, iter = 5000,
    chains = 4)
sim1_linear_result <- extract(sim1_linear, c("alpha"))
sim1_linear_mode <- postmode(sim1_linear_result$alpha)
sim1_linear_mode / sum(sim1_linear_mode)

# 4.1.2 Sim1 Comp
set.seed(888)
sim1_comp <- stan_solve(obs, "compmodel.stan", 5000)
sim1_comp_result <- extract(sim1_comp, c("xi"))
sim1_comp_mode <- postmode(sim1_comp_result$xi)
sim1_comp_mode / sum(sim1_comp_mode)

# 4.2 Simulation 2
obs_data <- c(  NA,   .6,   .3,   NA,   .1,   NA,
                .5,   NA,   .2,   .3,   NA,   NA,
                NA,   NA,   NA,   .5,   .25,  .25,
                NA,   .45,  .3,   NA,   .15,  .10,
                .5,   .5,   NA,   NA,   NA,   NA,
                .6,   NA,   NA,   .35,  .05,  NA)
obs <- t(matrix(obs_data, 6, 6))

# 4.2.1 sim2 Linear
scores <- melt(t(obs))
colnames(scores) <- c("rated", "rater", "S")
scores <- scores[!is.na(scores$S),]
scores <- scores[,c(2,1,3)]
X <- t(matrix(as.integer(!is.na(obs_data)), 6,6))          # What observations exist
D <- 6
simul_dat <- list (
  N = length(scores$S),
  S = scores$S,
  rater = scores$rater,
  rated = scores$rated,
  D = D,
  X = X,
  ones <- rep(1,D)
)

set.seed(888)
sim2_linear <- stan(file =  "linearmodel.stan", data = simul_dat, iter = 5000,
    chains = 4)
sim2_linear_result <- extract(sim2_linear, c("alpha"))
sim2_linear_mode <- postmode(sim2_linear_result$alpha)
sim2_linear_mode / sum(sim2_linear_mode)

# 4.1.2 sim2 Comp
set.seed(888)
sim2_comp <- stan_solve(obs, "compmodel.stan", 5000)
sim2_comp_result <- extract(sim2_comp, c("xi"))
sim2_comp_mode <- postmode(sim2_comp_result$xi)
sim2_comp_mode / sum(sim2_comp_mode)

save(sim1_linear, sim1_comp, sim2_linear, sim2_comp, file="Sim12Results.Rdata")
```

Simulations 3 and 4

```
# Script File
# Fee Division Simulation 3 & 4
# Edward K. Cheng

# Initialization
rm(list = ls())    # clear all

# Local Specific Commands
setwd("/Users/ekcheng/Documents/Dissertation/Chapter2/Code")
library(rstan)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
library(stats)
library(foreign)
library(MASS)
library(reshape)

set.seed(3.1415)

# 1. FUNCTIONS
# Both alr and alr.inv are checked

alr <- function(x) {
  d <- dim(x)
  if (d[1] != d[2]) print("WARNING (alr): Data matrix not square")
  N <- d[1]
  n <- N-1
  x_N <- x[,N]

  r <- matrix(NA, N, n)          # Creating normalized r vector
  for (i in 1:N) {
    r[i,1:n] <- x[i,1:n] /x_N[i]
    }
  lr <- log(r)               # Take log

  lr
  }

alr.inv <- function(lr) {
  d <- dim(lr)
  if (d[1] != d[2] + 1) print("WARNING (alr): Transformed matrix not N x n")
  N <- d[1]
  n <- d[2]

  r <- exp(lr)
  x <- matrix(NA, N, N)
  x[,N] <- 1 / (1 + rowSums(r))
  for (i in 1:N) {
    x[i,1:n] <- r[i,1:n] * x[i,N]
    }
  x
  }

alr.single <- function(x) {
  N <- length(x)
  n <- N-1
  r <- x[1:n] /x[N]
  lr <- log(r)               # Take log
  lr
}

alr.inv.single <- function(lr) {
  n <- length(lr)
  N <- n+1

  r <- exp(lr)
```

```
  x <- rep(NA, N)
  x[N] <- 1 / (1 + sum(r))
  x[1:n] <- r * x[N]
  x
  }

Fmatrix <- function (d) {
  cbind(diag(d), rep(-1,d))
}

Hmatrix <- function (d) {
  diag(d) + matrix(1,nrow=d,ncol=d)
}

Zmatrix <- function (scorevector) {
  w <- !is.na(scorevector)      # vector of non-missing values
  D <- length(scorevector)      # dimension of full simplex
  I <- diag(D)
  I[(w==1),]
}

getC <- function (scorevector) {
  C <- sum(!is.na(scorevector))
}

## Q Matrix -- Aitchison p.119
## Test by feeding it a basic vector of scores and see if it gives you the reduction matrix
Qmatrix <- function(scorevector) {
  w <- !is.na(scorevector)      # vector of non-missing values
  D <- length(scorevector)      # dimension of full simplex
  C <- sum(w)          # dimension of subcomposition simplex
  Z <- Zmatrix(scorevector)

  Fmatrix(C-1) %*% Z %*% t(Fmatrix(D-1)) %*% solve(Hmatrix(D-1))     # Results in correct result on test
}

# Normalization function
normalize <- function (scorevector) {
  scorevector / sum(scorevector[!is.na(scorevector)])
}

# 2. DATA GENERATION
set.seed(888)

s_true <- c(0.25, 0.15, 0.15, 0.1, 0.1, 0.1, 0.05, 0.05, 0.025, 0.025)
N <- length(s_true)

# Creates N observations where each observation has independent error (not normalized)
# Builds by column, so you can't combine with next for loop
s_obs_full <- NULL
for (i in 1:N){
  s_obs_full <- cbind(s_obs_full, rnorm(N, mean=s_true[i], sd = 0.2*s_true[i]))
}

s_true_err <- diag(s_obs_full)

s_base <- matrix(NA, N, N)
s_selfA <- matrix(NA, N, N)
s_selfB <- matrix(NA, N, N)
s_selfC <- matrix(NA, N, N)
s_selfD <- matrix(NA, N, N)
s_colludeA <- matrix(NA, N, N)

s_colludeC <- matrix(NA, N, N)
s_colludeD <- matrix(NA, N, N)
```

```
for (i in 1:N) {
    obs <- s_true + rnorm(N, mean=rep(0, N), sd = 0.2*s_true)      # Observations with error

    maxobs <- ceiling(N * s_true[i]/max(s_true))    # Max number of firms observed is the relative amount of work times N
    #minobs <- max(c(2, maxobs/2))                  # Min number of firms observed is the greater of 2 or max number divided by 2
    obs_sample <- sample(c(1:N)[-i], max(2,min(maxobs,N-1)), replace=F, prob=s_true[-i])     # Creates random number of observations (related to s_true), randomly selected

              # Don't allow self-report (handled later)

               # Weighted so that large shares are more observed

    s_base[i,obs_sample] <- obs[obs_sample]
    s_base[i,i] <- NA                               # No self-reporting allowed
    s_base[i,] <- normalize(s_base[i,])

    # True self-report
    s_selfA[i,obs_sample] <- obs[obs_sample]
    s_selfA[i,i] <- s_true_err[i]                      # Note -- will be normalized
    s_selfA[i,] <- normalize(s_selfA[i,])

    # Everyone overrates themselves
    s_selfB[i,obs_sample] <- obs[obs_sample]
    s_selfB[i,i] <- s_true[i] * 2                 # Note -- will be normalized
    s_selfB[i,] <- normalize(s_selfB[i,])

    # Only Firm 3 overrates itself
    s_selfC[i,obs_sample] <- obs[obs_sample]
    s_selfC[i,i] <- s_true_err[i]                      # Note -- will be normalized
    if (i==3) s_selfC[i,3] <- s_true[3] * 2
    s_selfC[i,] <- normalize(s_selfC[i,])

    # Only Firm 9 overrates itself
    s_selfD[i,obs_sample] <- obs[obs_sample]
    s_selfD[i,i] <- s_true_err[i]                      # Note -- will be normalized
    if (i==9) s_selfD[i,9] <- s_true[9] * 4
    s_selfD[i,] <- normalize(s_selfD[i,])


    # No self-report, Firms 4 and 5 collude
    s_colludeA[i,obs_sample] <- obs[obs_sample]
    s_colludeA[i,i] <- NA                      # Note -- will be normalized
    if (i==4) s_colludeA[i,5] <- s_true[5] * 3
    if (i==5) s_colludeA[i,4] <- s_true[4] * 4
    s_colludeA[i,] <- normalize(s_colludeA[i,])

    # There is NO colludeB for now

    # True self-report, Firms 4 and 5 collude
    s_colludeC[i,obs_sample] <- obs[obs_sample]
    s_colludeC[i,i] <- s_true_err[i]                      # Note -- will be normalized
    if (i==4) s_colludeC[i,4] <- s_true[4] * 4
    if (i==5) s_colludeC[i,5] <- s_true[5] * 3
    if (i==4) s_colludeC[i,5] <- s_true[5] * 3
    if (i==5) s_colludeC[i,4] <- s_true[4] * 4
    s_colludeC[i,] <- normalize(s_colludeC[i,])

    # Firm 3 overrates and Firms 4 and 5 collude
    s_colludeD[i,obs_sample] <- obs[obs_sample]
    s_colludeD[i,i] <- s_true_err[i]                      # Note -- will be normalized
    if (i==3) s_colludeD[i,3] <- s_true[3] * 2
    if (i==4) s_colludeD[i,4] <- s_true[4] * 3
    if (i==5) s_colludeD[i,5] <- s_true[5] * 2
```

```
    if (i==4) s_colludeD[i,5] <- s_true[5] * 2
    if (i==5) s_colludeD[i,4] <- s_true[4] * 3
    s_colludeD[i,] <- normalize(s_colludeD[i,])

}

write.csv(s_base, "Data7Base.csv")
write.csv(s_selfA, "Data7SelfA.csv")
write.csv(s_selfB, "Data7SelfB.csv")
write.csv(s_selfC, "Data7SelfC.csv")
write.csv(s_selfD, "Data7SelfD.csv")
write.csv(s_colludeA, "Data7colludeA.csv")
write.csv(s_colludeC, "Data7colludeC.csv")
write.csv(s_colludeD, "Data7colludeD.csv")

# 3 SOLVE

# 3.1 STAN EXECUTION FUNCTION

stan_solve <- function(S, stancode) {
# Create Q and Z matricies -- to handle missing data
  F <- nrow(S)
  N <- ncol(S)

  Qindex <- rep(NA, F)
  #Qlist <- NULL
  Q <- NULL

  Zindex <- rep(NA, F)
  Zlist <- NULL
  Z <- NULL

  C <- rep(NA, F)

  # Loop
  for (i in 1:F) {
    newQ <- Qmatrix(S[i,])          # Round very small values
    newZ <- Zmatrix(S[i,])

    Q <- rbind(Q, newQ)
    Z <- rbind(Z, newZ)
    C[i] <- getC(S[i,])

    #Qlist[[i]] <- newQ                          # Test code
    Zlist[[i]] <- newZ

    if (i==1) {                                # Sets the i=1
      Qindex[i] <- 1
      Zindex[i] <- 1
    }
    if (i < F) {                           # Sets the next index
      Qindex[i+1] <- Qindex[i] + C[i] - 1
      Zindex[i+1] <- Zindex[i] + C[i]
    }
  }

  S_all <- S                  # Do this to ease transfer of data to STAN
  S_all[is.na(S)] <- -99       # Missing values are tagged with -99 so we can detect errors in STAN

  simul_dat <- list (
    N = N,
    F = F,
    S_all = S_all,
    rater = c(1:F),

    Q_all = Q,
    Qindex = Qindex,
```

```
    Qlength = nrow(Q),

    Z_all = Z,
    Zindex = Zindex,
    Zlength = nrow(Z),

    C_all = C
  )

  ptm <- proc.time()
  stan_fit <- stan(file = stancode, data = simul_dat,
                   iter = 1000, chains = 1)
  proc.time() - ptm

  stan_fit
}

# 4.2 Runs
fit_base_no <- stan_solve(s_base, "compmodel4.stan")
fit_selfA_no <- stan_solve(s_selfA, "compmodel4.stan")
fit_selfB_no <- stan_solve(s_selfB, "compmodel4.stan")

fit_base_resist <- stan_solve(s_base, "compmodel5.stan")
fit_selfA_resist <- stan_solve(s_selfA, "compmodel5.stan")
fit_selfB_resist <- stan_solve(s_selfB, "compmodel5.stan")

fit_selfC_no <- stan_solve(s_selfC, "compmodel4.stan")
fit_selfD_no <- stan_solve(s_selfD, "compmodel4.stan")
fit_selfC_resist <- stan_solve(s_selfC, "compmodel5.stan")
fit_selfD_resist <- stan_solve(s_selfD, "compmodel5.stan")

fit_colludeA_no <- stan_solve(s_colludeA, "compmodel4.stan")
fit_colludeC_no <- stan_solve(s_colludeC, "compmodel4.stan")
fit_colludeD_no <- stan_solve(s_colludeD, "compmodel4.stan")

fit_colludeA_resist <- stan_solve(s_colludeA, "compmodel5.stan")
fit_colludeC_resist <- stan_solve(s_colludeC, "compmodel5.stan")
fit_colludeD_resist <- stan_solve(s_colludeD, "compmodel5.stan")

save(fit_base_no, fit_selfA_no, fit_selfB_no, fit_selfC_no, fit_selfD_no, fit_
    colludeA_no, fit_colludeC_no, fit_colludeD_no,
      fit_base_resist, fit_selfA_resist, fit_selfB_resist, fit_selfC_resist, fit_
    selfD_resist, fit_colludeA_resist, fit_colludeC_resist, fit_colludeD_resist,
      file="Results7.Rdata")

# 5. ANALYSIS (using draws to calculate posterior mean and mode)
load("Results7.Rdata")

postmode <- function (draws) {
    N <- ncol(draws)
    estmode <- rep(NA, N)
    for (i in 1:N) {
      dens <- density(draws[,i])
      estmode[i] <- dens$x[dens$y == max(dens$y)]
    }
    estmode
}

draws_base_no <- extract(fit_base_no, c("xi", "Omega", "sigma"))
postmode(draws_base_no$xi)

draws_selfA_no <- extract(fit_selfA_no, c("xi", "Omega", "sigma"))
postmode(draws_selfA_no$xi)

draws_selfB_no <- extract(fit_selfB_no, c("xi", "Omega", "sigma"))
postmode(draws_selfB_no$xi)

draws_selfC_no <- extract(fit_selfC_no, c("xi", "Omega", "sigma"))
```

```
postmode(draws_selfC_no$xi)

draws_selfD_no <- extract(fit_selfD_no, c("xi", "Omega", "sigma"))
postmode(draws_selfD_no$xi)

draws_colludeA_no <- extract(fit_colludeA_no, c("xi", "Omega", "sigma"))
postmode(draws_colludeA_no$xi)

draws_colludeC_no <- extract(fit_colludeC_no, c("xi", "Omega", "sigma"))
postmode(draws_colludeC_no$xi)

draws_colludeD_no <- extract(fit_colludeD_no, c("xi", "Omega", "sigma"))
postmode(draws_colludeD_no$xi)

draws_base_resist <- extract(fit_base_resist, c("xi", "Omega", "sigma", "gamma"))
postmode(draws_base_resist$xi)

draws_selfA_resist <- extract(fit_selfA_resist, c("xi", "Omega", "sigma", "gamma"))
postmode(draws_selfA_resist$xi)

draws_selfB_resist <- extract(fit_selfB_resist, c("xi", "Omega", "sigma", "gamma"))
postmode(draws_selfB_resist$xi)

draws_selfC_resist <- extract(fit_selfC_resist, c("xi", "Omega", "sigma", "gamma"))
postmode(draws_selfC_resist$xi)

draws_selfD_resist <- extract(fit_selfD_resist, c("xi", "Omega", "sigma", "gamma"))
postmode(draws_selfD_resist$xi)

draws_colludeA_resist <- extract(fit_colludeA_resist, c("xi", "Omega", "sigma", "
    gamma"))
postmode(draws_colludeA_resist$xi)

draws_colludeC_resist <- extract(fit_colludeC_resist, c("xi", "Omega", "sigma", "
    gamma"))
postmode(draws_colludeC_resist$xi)

draws_colludeD_resist <- extract(fit_colludeD_resist, c("xi", "Omega", "sigma", "
    gamma"))
postmode(draws_colludeD_resist$xi)

comp7results <- data.frame(
  truth = s_true,
  base_no = postmode(draws_base_no$xi),
  selfA_no = postmode(draws_selfA_no$xi),
  selfB_no = postmode(draws_selfB_no$xi),
  selfC_no = postmode(draws_selfC_no$xi),
  selfD_no = postmode(draws_selfD_no$xi),
  colludeA_no = postmode(draws_colludeC_no$xi),
  colludeC_no = postmode(draws_colludeC_no$xi),
  colludeD_no = postmode(draws_colludeD_no$xi),
  base_resist = postmode(draws_base_resist$xi),
  selfA_resist = postmode(draws_selfA_resist$xi),
  selfB_resist = postmode(draws_selfB_resist$xi),
  selfC_resist = postmode(draws_selfC_resist$xi),
  selfD_resist = postmode(draws_selfD_resist$xi),
  colludeA_resist = postmode(draws_colludeC_resist$xi),
  colludeC_resist = postmode(draws_colludeC_resist$xi),
  colludeD_resist = postmode(draws_colludeD_resist$xi))

comp7norm <- comp7results
for(i in 1:ncol(comp7norm)) {
    comp7norm[,i] <- comp7norm[,i] / sum(comp7norm[,i])
}
```

Stan Code

Linear Model

```
data {
  int<lower=1> D;                  // number of firms
  int<lower=1> N;                  // number of observations
  real<lower=0, upper=1> S[N];          // scores given by firm i to firm j
  int<lower=1, upper=D> rater[N];         // array of rater variable
  int<lower=1, upper=D> rated[N];       // array of target variable
  row_vector<lower=0, upper=1>[D] X[D];    // presence of scores from firm i to firm j
  vector[D] ones;                  // vector of ones
  }

parameters {
  simplex[D] alpha;                // variable of interest -- value of firm to litigation
  real sigma[D];                     // Error variation
  }

model {
// priors
  alpha ~ dirichlet(ones);            // Dirichlet(1) is supposed to be uniform distribution

// draws
  for (i in 1:D) {
    sigma[i] ~ inv_gamma(0.001,0.001);
    }

  for (i in 1:N) {
    S[i] ~ normal(alpha[rated[i]] / dot_product(alpha, X[rater[i]]), sigma[rater[i]]);
    }
}
```

Compositional Model

```
data {
  int<lower=1> N;                  // number of rated firms
  int<lower=1> F;                  // number of rating firms
  int<lower=1> Qlength;            // nrow of Q
  int<lower=1> Zlength;            // nrow of Z
  matrix[N,N] S_all;                 // Observed Raw Compositions (no log ratio yet)
  int<lower=1> rater[F];               // Rater number
  int<lower=1, upper=Qlength> Qindex[F];       // Indexes for Q and Z
  int<lower=1, upper=Zlength> Zindex[F];
    matrix<lower=-2, upper=2>[Qlength, N-1] Q_all;     // Mashed Q matrix  Should be -1 to 1                          //     roundoff error
    matrix<lower=0, upper=1>[Zlength, N]   Z_all;      // Mashed Z matrix
    int<lower=1> C_all[F];            // C vector -- contains lengths for each set of scores
}

transformed data {
    int<lower=1> n;
    n = N-1;
}

parameters {
  vector[n] alpha;                 // variable of interest -- true composition vector
  cholesky_factor_corr[n] Omega;
    vector<lower=0>[n] sigma;
  }
```

```
model {
  matrix[n, n] Sigma;

  Sigma = diag_pre_multiply(sigma, Omega);

  to_vector(alpha) ~ normal(0, 5);
  Omega ~ lkj_corr_cholesky(4);
  sigma ~ cauchy(0, 2.5);


  // First iteration
  for (i in 1:F) {
    int C = C_all[i];
    int c = C-1;
    vector[C] psi;
    vector[c] alrS;
    matrix[c,n] Q;

    psi = block(Z_all, Zindex[i], 1, C, N) * row(S_all, i)';
    alrS = log( head(psi,c) / psi[C] );

    Q = block(Q_all, Qindex[i], 1, c, n);

    alrS ~ multi_normal_cholesky(Q*alpha, Q*Sigma*Q');
  }
}

generated quantities {
  real<lower=0,upper=1> xi[N];

  xi[N] = 1 / (1 + sum(exp(alpha)));
  for (i in 1:n) {
    xi[i] = exp(alpha[i]) * xi[N];
  }
}
```

Compositional Model with Collusion Resistance

```
data {
  int<lower=1> N;                // number of rated firms
  int<lower=1> F;                // number of rating firms
  int<lower=1> Qlength;          // nrow of Q
  int<lower=1> Zlength;          // nrow of Z
  matrix[N,N] S_all;               // Observed Raw Compositions (no log ratio yet)
  int<lower=1> rater[F];             // Rater number
  int<lower=1, upper=Qlength> Qindex[F];      // Indexes for Q and Z
  int<lower=1, upper=Zlength> Zindex[F];
  matrix<lower=-2, upper=2>[Qlength, N-1] Q_all;     // Mashed Q matrix  Should be -1
     to 1, but there was some
       //     roundoff error
    matrix<lower=0, upper=1>[Zlength, N]  Z_all;      // Mashed Z matrix
  int<lower=1> C_all[F];            // C vector -- contains lengths for each set of
     scores
}

transformed data {
    int<lower=1> n;
    n = N-1;
}

parameters {
  vector[n] alpha;               // variable of interest -- true composition vector
  cholesky_factor_corr[n] Omega;
    vector<lower=0, upper=1>[n] sigma;
    real<lower=0> gamma[F];                 // random effects variable
    real<lower=0> sig_gamma;
  }
```

```
model {
  matrix[n, n] Sigma;

  to_vector(alpha) ~ normal(0, 5);
  Omega ~ lkj_corr_cholesky(4);
  sigma ~ cauchy(0, 2.5);
  gamma ~ normal(0, sig_gamma);

  // First iteration
  for (i in 1:F) {
    int C = C_all[i];
    int c = C-1;
    vector[C] psi;
    vector[c] alrS;
    matrix[c,n] Q;

    Sigma = diag_pre_multiply(sigma * gamma[i], Omega);

    psi = block(Z_all, Zindex[i], 1, C, N) * row(S_all, i)';
    alrS = log( head(psi,c) / psi[C] );

    Q = block(Q_all, Qindex[i], 1, c, n);

    alrS ~ multi_normal_cholesky(Q*alpha, Q*Sigma*Q');
  }
}

generated quantities {
  real<lower=0,upper=1> xi[N];

  xi[N] = 1 / (1 + sum(exp(alpha)));
  for (i in 1:n) {
    xi[i] = exp(alpha[i]) * xi[N];
  }
}
```

Code for Chapter 3 (Latent Space)

R Code

Primary Script

```
# Initialization
rm(list = ls())    # clear all
filename = "latent7d"
series = "latent7"

# HPC Specific Commands
.libPaths("/vega/stats/users/ekc2109/rpackages/")   # Use for cluster

library(rstan)
set_cppo("fast")
library(stats)
library(foreign)
library(MASS)
library(graphics)
library(matrixcalc)
library(igraph)
library(gdata)
library(shapes)
library(reshape)

# 1. FUNCTIONS


# 2. LOAD DATA AND PREP CITATIONS

# 2.01 Justice centered votes
load("SCJD.Rdata")
SCJD <- SCDB_2013_01_justiceCentered_Citation
SCJD <- SCJD[SCJD$term < 2003, ]        # Eliminate cases beyond 2002 for now

# 2.02 SCJD Case-Level Data
SCJDcases <- read.csv("SCJDcasesEKC.csv")         # Load dataset with short names
    for 1st Am
SCJDcases <- SCJDcases[SCJDcases$term < 2003, ]   # Eliminate cases beyond 2002 for
    now

# 2.03 Citation Network (Fowler)
casenet <- read.csv("SCcitation.csv")

# 2.04 Citations for 2002 test set
# These are the citations made by the lower court opinion, which is all you will
    have in actual prediction
edge2002 <- read.csv("2002edges.csv", header=FALSE)
colnames(edge2002) <- c("citer", "cited", "weight")

# 2.05 Justice Biographical Data
# Read in RData format because the excel conversion takes too long
#justicesbio <- read.xlsx("justicesbio.xlsx", 1)  # Code for setting up the R data
    format
#save(justicesbio, file="JusticesBio.Rdata")
load("JusticesBio.Rdata")

# 2.1 Construct lookup table
caselookup <- data.frame(usCite=SCJDcases$usCite, lexisCite=SCJDcases$lexisCite,
    name=SCJDcases$shortName)
caselookup <- caselookup[!is.na(caselookup$lexisCite),]
caselookup$usCite <- gsub("[\\.\\s]","", caselookup$usCite, perl=TRUE)      # Remove
     periods and spacing to match the citation parser
lexissplit <- strsplit(as.character(caselookup$lexisCite), " U.S. LEXIS ")

```

```r
newlexis <- character(length(caselookup$lexisCite))
for (i in 1:length(newlexis)) {
  year <- unlist(lexissplit[[i]])[1]
  num <- unlist(lexissplit[[i]])[2]
  num <- sprintf("%04d", as.integer(num))   # Pad with zeroes
  newlexis[i] <- paste(year, "USLX", num, sep="")
  }

# test <- data.frame(caselookup$lexisCite, newlexis)
caselookup$newlexis <- newlexis;

# 2.2 2002 Lookup Table for research
#lookup2002 <- data.frame(usCite=SCJDcases$usCite, lexisCite=SCJDcases$lexisCite,
    caseName=SCJDcases$caseName, term=SCJDcases$term)
#lookup2002 <- lookup2002[lookup2002$term == 2002, ]
# write.csv(lookup2002, "2002cases.csv")

# 2.3 Convert the 2002 citations
# Produce list with 2002 Supreme Court Lexis Cite and the opinions cited by the
    lower court
index <- match(as.character(edge2002$cited), caselookup$usCite)
edge2002$uscited <- edge2002$cited
edge2002$cited <- caselookup$newlexis[index]

lookup2002 <- read.csv("2002lookup.csv")
lookup2002 <- data.frame(supctlexis = lookup2002$supctlexis, lowerct=lookup2002$
    lowerct)

edge2002$citersc <- numeric(nrow(edge2002))
lookup2002$lowerct <- as.character(lookup2002$lowerct)
edge2002$citer <-as.character(edge2002$citer)
for (i in 1:nrow(edge2002)) {
  if (edge2002$citer[i] %in% lookup2002$lowerct) {
    edge2002$citersc[i] <- as.character(lookup2002$supctlexis[which(lookup2002$
    lowerct == edge2002$citer[i], arr.in=TRUE)])
    }
  else { edge2002$citersc[i] <- NA }
  }

edge2002 <- edge2002[!is.na(edge2002$cited),]     # Just keep the ones with unique
    Lexis cite

# 2.4 Justice lookup table
# Don't use the SCJD lookup because that starts with an offset
# justicelookup <- data.frame(SCJDcode = SCJD$justice, name = SCJD$justiceName)
# unique(justicelookup)

# 2.5 Convert and Reduce the Fowler Case Network
index_citer <- match(as.character(casenet$citing_case), caselookup$lexisCite)
casenet$citer <- caselookup$newlexis[index_citer]

index_cited <- match(as.character(casenet$cited_case), caselookup$lexisCite)
casenet$cited <- caselookup$newlexis[index_cited]

# Note: Because you used a lookup table limited by SCJD to construct citer and cited
    , you
# Can use the NA as the filter out cases before the 1946 cutoff
# When you move to more expansive networks, will have to revisit
casenet <- casenet[!is.na(casenet$citer),]
casenet <- casenet[!is.na(casenet$cited),]
casenet <- casenet[,3:4]

# 2.6 SCJD Conversion
index_case <- match(as.character(SCJD$lexisCite), caselookup$lexisCite)
SCJD$newlexisCite <- caselookup$newlexis[index_case]
# data.frame(SCJD$lexisCite, SCJD$newlexisCite)

index_case <- match(as.character(SCJDcases$lexisCite), caselookup$lexisCite)
```

```
SCJDcases$newlexisCite <- caselookup$newlexis[index_case]

# 3. Construct datasets for model
# Supreme Court Database is 1946 forward.
# SCJD$vote = concur/dissent
# 3.1 Vote table

votelist <- data.frame(justice=SCJD$justiceName, citation=SCJD$newlexisCite, issueArea=SCJD$issueArea, term=SCJD$term,
  direction=SCJD$direction, origvote=SCJD$vote, voteUnclear=SCJD$voteUnclear,
  disp=SCJD$caseDisposition, petwin=SCJD$partyWinning, spaethid=SCJD$justice,
  issue=SCJD$issue, lcDisagreement=SCJD$lcDisagreement, lcDisposition=SCJD$lcDisposition,
  lcDispositionDirection=SCJD$lcDispositionDirection, lawSupp=SCJD$lawSupp)

# 3.1.1 Create Consolidated Disposition Variable
votelist$simpledisp <- rep(NA, nrow(votelist))
votelist$simpledisp[votelist$disp == 2] <- 0  # Affirm
votelist$simpledisp[votelist$disp == 3] <- 1  # Reverse
votelist$simpledisp[votelist$disp == 4] <- 1  # R&R
votelist$simpledisp[votelist$disp == 5] <- 1  # Vacate and remand
votelist$simpledisp[votelist$disp == 6] <- 1  # Reverse in part
votelist$simpledisp[votelist$disp == 7] <- 1  # Reverse in part and remand
# NA for the remainder, i.e., 1(petition granted), 8(vacated), 9(appeal dismissed), 10(certification), 11 (no disposition)

# 3.1.2 Winning Variable
# For preliminaries, use Winning Party variable rather than 3.1.1 Consolidated Disposition
# 0 = affirm (not favorable to petitioner), 1 = reverse-like (favorable to petitioner), 2 = unclear

# 3.1.3 Create Join Variable
votelist$join <- rep(NA, nrow(votelist))
votelist$join[votelist$origvote==1] <- 1    # Join majority
votelist$join[votelist$origvote==2] <- 0    # Dissent
votelist$join[votelist$origvote==3] <- 1    # Regular Concurrence (see Codebook 57: This is very close to join)
votelist$join[votelist$origvote==4] <- 1    # Special Concurrence (see Codebook 57: This is concurring in judgment)
votelist$join <- as.logical(votelist$join)
# NA for the remainder, i.e., 5(judgment), 6(dissent from cert), 7(juris dissent), 8(equal divided)

# 3.1.4 Create New Vote Variable
votelist$petwin[votelist$petwin == 2] <- NA   # Cases of unclear disposition
votelist$petwin <- as.logical(votelist$petwin)

votelist$vote <- (votelist$petwin == votelist$join)   # See Notes -- a truth table bears this out
votelist$vote[is.na(votelist$petwin)] <- NA       # Handle the NAs
votelist$vote[is.na(votelist$join)] <- NA

votelist$vote <- as.logical(votelist[,'vote'])      # Use of "$" risks partial match
votelist$vote <- !votelist$vote              # IMPORTANT: Flip so that 1=affirm(agree) and 0=reverse(disagree)
# data.frame(votelist$petwin, votelist$join, votelist$vote)

# 3.1.5 Remove NA votes
votelist <- votelist[!is.na(votelist$vote),]

# 3.1.6 Standardize the ideology variable
# SCJD Direction = Conservative (1) or Liberal (2) or Unclear (3)
# Make ideology = Conservative (0) or Liberal (1) or Unclear (NA)
votelist$ideovote <- NA
votelist$ideovote[votelist$direction=="1"] <- 0
votelist$ideovote[votelist$direction=="2"] <- 1

```

```r
# 3.2 Justice Dataset

# 3.2.1 Clean Data
# Variables of interest
# name, spaethid (matches SCJD), parnom  (political party of nominee), prespart (party of nominating President), ideo (Segal Cover score of nominee ideology)
# Note: Lots of other ideology scores available in the justice dataset, but right now, we only want baseline

justicelist <- data.frame(name=justicesbio$name, spaethid=justicesbio$spaethid, ideo=justicesbio$ideo, parnom=justicesbio$parnom)
justicelist <- unique(justicelist)    # Remove duplicate rows (artifact of dataset)

justicelist <- justicelist[justicelist$ideo != 777,]
justicelist <- justicelist[justicelist$spaethid != "no spaet",]
justicelist$ideo[justicelist$ideo == "data unav"] <- NA

justicelist$spaethid <- as.numeric(as.character(justicelist$spaethid))
justicelist$ideo <- as.numeric(as.character(justicelist$ideo))
justicelist <- justicelist[(justicelist$spaethid!=74 | !is.na(justicelist$ideo) ),]     # Get rid of duplicate for Stone, CJ (who was elevated)
justicelist <- justicelist[(justicelist$spaethid!=97 | justicelist$ideo!=0.845  ),]     # Get rid of duplicate for Fortas's CJ nomination

justicelist <- justicelist[order(justicelist$spaethid),]    # Put everything in order
rownames(justicelist) <- seq(length=nrow(justicelist))      # Reset rownames

# 3.2.2 Create party variable
# Does not capture Federalists, Indep, or Whigs, but those are not important for study
# Independents should be NA anyway
justicelist$party <- NA
justicelist$party[justicelist$parnom == "democrat"] <- 1
justicelist$party[justicelist$parnom == "republic"] <- 0

# 3.3 Add Justice Dataset to VoteList
votelist$party <- justicelist$party[votelist$spaethid]
votelist$ideojustice <- justicelist$ideo[votelist$spaethid]

# 4. Dataset Manipulations

# 4.1 Restrict Subject Matter
# Restrictions on issue
# 1: Crim Pro (1515 cases)
# 3: First Amendment (579 cases)
#
#
votedata <- votelist[(votelist$issueArea == 3)  ,]     # First Amendment

votedata$citation <- factor(as.character(votedata$citation))    # Reset the cases (get rid of cases not in set)
votedata$justice <- factor(as.character(votedata$justice))      # Reset the justices (get rid of justices not in set)

# 4.2 Recode Issue Codes
votedata$simpleissue <- (as.integer(votedata$issue) - 30000) / 10;    # Reduces issue codes to 1,2,3 index
votedata$simpleissue <- factor(as.character(votedata$simpleissue))      # Reset the justices (get rid of justices not in set)

# 4.3 Lower Court Ideology
votedata$lowerideo <- NA
votedata$lowerideo[votedata$lcDispositionDirection=="1"] <- 0
votedata$lowerideo[votedata$lcDispositionDirection=="2"] <- 1

# 4.4 Remove NAs
votedata <- votedata[!is.na(votedata$vote),]
```

```r

# 5. Divide Training and Holdout Sets / Store the Full Set
mask <- (votedata$term > 2001)      # Only 5 holdout cases in 1st Amendment
votedata_tr <- votedata[!mask,]
votedata_tr$citation <- factor(as.character(votedata_tr$citation))    # Reset the factor

votedata_ho <- votedata[mask,]                  # Save the actual Supreme Court votes for later use
votedata_ho$citation <- factor(votedata_ho$citation)    # Reset the factor

votedata_full <- votedata

# 6. Social Network Issues
# DIFFERENT FROM PREVIOUS
# Construct a combined citation set, with the test year consisting of lower court citations
traincases <- levels(votedata_tr$citation)
testcases <- levels(factor(edge2002$citersc))
testcases <- testcases[testcases %in% votedata_ho$citation] # Keep only cases in subject area

trainnet <- casenet[casenet$citer %in% traincases, ]      # Keep only training cases
testnet <- data.frame(citer=edge2002$citersc, cited=edge2002$cited)
testnet <- testnet[testnet$citer %in% testcases, ]

combnet <- rbind(trainnet, testnet)             # Training cases citations and Lower court citations

# 6.1 Reduce Networks to Needed Data
casesused <- c(traincases, testcases)

combnet[,1] <- as.character(combnet[,1])
combnet[,2] <- as.character(combnet[,2])

keeps <- apply(combnet[,1:2], 1, function(x) all(x %in% casesused))
  # Create keep list -- any rows in which citer and cited are in the casesused list

combnet.edge <- combnet[keeps,]
colnames(combnet.edge) <- c("citer", "cited")

# 6.2 Build adjacency matrix
combnet.net <- graph.data.frame(combnet.edge, directed=TRUE, vertices=data.frame(casesused))      # Ensure that network has all cases as verticies
combnet.adj <- get.adjacency(combnet.net, sparse=TRUE)
combnet.adj <- combnet.adj[sort(rownames(combnet.adj)),]
combnet.adj <- combnet.adj[,sort(colnames(combnet.adj))]

# 6.3 Check lower triangular
# The adjacency matrix is not perfectly lower triangular because of citation between cases from same day

#all(upperTriangle(casenet.adj) == 0)      # Not completely upper triangular
test <- combnet.adj * upper.tri(combnet.adj)  # Mask off all of the lower triangle
                           # Look only at the violations
  # all(lowerTriangle(test) == 0)        # Sanity check

which(as.matrix(test)!= 0, arr.in=TRUE)          # The violations are in cases very close to each other

# 6.3.1 Remove offending cases
# For now, delete the offending citations
# You could reorder the matrix to deal with this problem, but it's too confusing
# And if you are predicting within the year, you won't see these anyway

combnet.adj <- combnet.adj * lower.tri(combnet.adj)
#all(upperTriangle(combnet.adj) == 0)   # Verify that it is now lower triangular
```

```
# 6.3.2 Make sure adjacency matrix is binary
# which(as.matrix(combnet.adj) > 1, arr.in=TRUE)      # Small number of variations (don't know why)
combnet.adj[which(as.matrix(combnet.adj) > 1, arr.in=TRUE)]

combnet.adj[combnet.adj != 0] <- 1      # Wipe them out

# 6.4 Separate into test and training groups
K <- length(traincases)
K_til <- length(testcases)
combnet_adj_tr <- combnet.adj[1:K, 1:K]
combnet_adj_ho <- combnet.adj[(K+1):(K+K_til), 1:K]

# 7. Case-Level Data
# 7.1 Organize the Data
 caselist <- data.frame(citation=SCJDcases$newlexisCite, disp=SCJDcases$caseDisposition, term=SCJDcases$term,
   direction=SCJDcases$decisionDirection,lcDispositionDirection=SCJDcases$lcDispositionDirection, lawSupp=SCJDcases$lawSupp, issue=SCJDcases$issue )

 caselist$lowerideo <- NA
 caselist$lowerideo[caselist$lcDispositionDirection=="1"] <- 0
 caselist$lowerideo[caselist$lcDispositionDirection=="2"] <- 1

 caselist$ideovote <- NA
 caselist$ideovote[caselist$direction=="1"] <- 0
 caselist$ideovote[caselist$direction=="2"] <- 1

# 7.2 Do Lookup Action To Build Case Outcome Dataframe
 index_case_tr <- match(traincases, caselist$citation)
 casedata_tr <- caselist[index_case_tr,]

#data.frame(casedata_tr$citation, levels(votedata_tr$citation))   # Sanity Check
#all.equal(as.character(casedata_tr$citation), levels(votedata_tr$citation))

 index_case_ho <- match(testcases, caselist$citation)
 casedata_ho <- caselist[index_case_ho,]

#data.frame(casedata_ho$citation, levels(votedata_ho$citation))   # Sanity Check
#all.equal(as.character(casedata_ho$citation), levels(votedata_ho$citation))

# 7.3 Check that full dataset also works
casedata_full <- rbind(casedata_tr, casedata_ho)
data.frame(casedata_full$citation, levels(votedata_full$citation))    # Sanity Check
all.equal(as.character(casedata_full$citation), levels(votedata_full$citation))

# 8. Prepare Ideological Agreement Matrix
# Can't use this on testing set, unless we start inferring ideological agreement

# Select set you are using
casedata <- casedata_tr
votedata <- votedata_tr

make.agreematrix <- function(x) {
  C <- length(x)
  R <- matrix(NA, C, C)
  for (i in 1:C) {
    R[i,] <- as.integer(x == x[i])
    }
  R
  }

# 8.1 Agreement based on manual ideological coding  (Model 3)
 AgreeZideo <- make.agreematrix(casedata$ideovote)

# 8.2 Agreement based on judge joins
# votedata is a listing of all votes
agreedata <- data.frame(justice=votedata$justice, citation=votedata$citation, join=
```

```r
    votedata$join)

justices <- levels(agreedata$justice)
cases <- levels(agreedata$citation)
J <- length(justices)
K <- length(cases)

agreematrix <- array(data=NA, dim=c(J,K,K), dimnames=c("justice", "case1", "case2"))
for (j in 1:J) {
  joindata <- agreedata[agreedata$justice==justices[j],]        # Take just that
    justice's data
  joinvector <- rep(NA, K)
  for (i in 1:nrow(joindata)) {
    joinvector[as.integer(joindata$citation[i])] <- joindata$join[i]
    }
  agreematrix[j,,] <- make.agreematrix(joinvector)          # Create agreement
    matrix
  }

# 8.2.1 Count up judge agreement (Model 4)
# Recode
agreematrix4 <- agreematrix
agreematrix4[agreematrix4==0] <- -1
agreematrix4[is.na(agreematrix4)] <-0

AgreeZjoin <- colSums(agreematrix4)

# 8.2.2. Latent judge agreement model (Model 5)
# For the latent model, you don't want any NAs
# Just want the instances of data

AgreeZlatent <- data.frame(case1 = NA, case2 = NA, agree = NA)
tally <- 0

for (j in 1:J) {
  joindata <- agreedata[agreedata$justice==justices[j],]        # Take just that
    justice's data
  joinmatrix <- make.agreematrix(joindata$join)           # Create agreement matrix
  rownames(joinmatrix) <- joindata$citation
  colnames(joinmatrix) <- joindata$citation
  jointable <- melt(joinmatrix)                      # Melt matrix back to data.frame
    of Z_kl
  colnames(jointable) <- c("case1", "case2", "agree")

  AgreeZlatent <- rbind(AgreeZlatent, jointable)
  }

AgreeZlatent <- AgreeZlatent[-1,]                    # Remove NA placeholder
AgreeZlatent$case1 <- factor(AgreeZlatent$case1)
AgreeZlatent$case2 <- factor(AgreeZlatent$case2)

AgreeZlatent <- AgreeZlatent[as.character(AgreeZlatent$case1) > as.character(
    AgreeZlatent$case2),]

# 9. Prepare STAN (Phase 1)
# Note that when passing a matrix to STAN, you want to convert Boolean to numbers

# join = whether justice concurred with the majority opinion
# vote = whether justice voted to affirm lower court (1 = affirm, 0 = reverse)

agree_dat <- list (
          K = nrow(combnet_adj_tr),
          N_z = nrow(AgreeZlatent),
          AgreeZ = AgreeZlatent$agree,

          case1_z = as.numeric(AgreeZlatent$case1),
          case2_z = as.numeric(AgreeZlatent$case2)

```

```
        )

agree_fit <- stan(file = paste(filename, "_agree.stan", sep=""), init="random", data
    = agree_dat, iter = 1000, chains = 1)
agree_fit
# save(agree_fit, votedata_tr, votedata_ho, casedata_tr, casedata_ho, caselookup,
    file=paste(filename, "Agree.Rdata", sep=""))

# 10. Prepare STAN (Phase 2)

agree_results <- get_posterior_mean(agree_fit)

y <- agree_results[grep("y\\[", rownames(agree_results)),]    # Cut only part of
    table with coordinates
alpha_a <- agree_results[grep("alpha_a", rownames(agree_results)),]   # Cut only
    part of table with coordinates
cite_dat <- list (
        K = nrow(combnet_adj_tr),
        y = y,
        alpha_a = alpha_a,
        Citation = as.matrix(combnet_adj_tr)        # STAN doesn't recognize
    sparse matrix
        )

cite_fit <- stan(file = paste(filename, "_cite.stan", sep=""), init="random", data =
    cite_dat, iter = 1000, chains = 1)

cite_fit
#save(cite_fit, agree_fit, votedata_tr, votedata_ho, casedata_tr, casedata_ho,
    combnet_adj_tr, combnet_adj_ho, caselookup, file=paste(filename, ".Rdata", sep
    =""))


# 11. Procrustes Analysis
rawdata <- extract(cite_fit)
coords <- rawdata$x     # Bind both training and holdout sets for Procrustes
procdata <- aperm(coords, c(2,3,1))     # Permute matrix dimensions
procrustes.out <- procGPA(procdata)
#save(procrustes.out, cite_fit, agree_fit, votedata_tr, votedata_ho, casedata_tr,
    casedata_ho, combnet_adj_tr, combnet_adj_ho, caselookup, file=paste(filename, "
    Proc.Rdata", sep=""))

#12. PREDICTION MODEL

# 12.1 Get previously estimated data
# Saves time
#load("latent6c.Rdata")
#load("latent6cProc.Rdata")

cite_results <- get_posterior_mean(cite_fit)
agree_results <- get_posterior_mean(agree_fit)

alpha_c <- cite_results[grep("alpha_c", rownames(cite_results)),]
delta <- cite_results[grep("delta\\[", rownames(cite_results)),]
beta <- cite_results[grep("beta", rownames(cite_results)),]
x <- data.frame(x1=procrustes.out$mshape[,1], x2=procrustes.out$mshape[,2])
sigma_gamma <- cite_results[grep("sigma_gamma", rownames(cite_results)),]
sigma_x1 <- cite_results[grep("sigma_x1", rownames(cite_results)),]
sigma_x2 <- cite_results[grep("sigma_x2", rownames(cite_results)),]
rho_x <- cite_results[grep("rho_x", rownames(cite_results)),]

alpha_a <- agree_results[grep("alpha_a", rownames(agree_results)),]
y <- agree_results[grep("y\\[", rownames(agree_results)),]
sigma_y <- agree_results[grep("sigma_y", rownames(agree_results)),]


# 12.2 Justice case votes
# Actual Justice Votes on Test Set
```

```r
vote_actual <- data.frame(justice = votedata_ho$justice, citation = votedata_ho$citation, caseaffirm = !as.logical(votedata_ho$simpledisp),
        join = votedata_ho$join)
vote_actual$affirmvote <- !xor(vote_actual$caseaffirm, vote_actual$join)

# Justice Joins (Agreement) on Training Set
agreedata <- data.frame(justice=votedata_tr$justice, citation=votedata_tr$citation, join=votedata_tr$join)

Vtable <- cast(agreedata, justice ~ citation)
rownames(Vtable) <- Vtable[,1]
Vtable <- Vtable[,-1]

Vmatrix <- (as.matrix(Vtable))        # Convert away from list
rownames(Vmatrix) <- rownames(Vtable)
colnames(Vmatrix) <- colnames(Vtable)

Vmatrix[Vmatrix==TRUE] <- 1
Vmatrix[Vmatrix==FALSE] <- -1
Vmatrix[is.na(Vmatrix)] <- 0

# 12.3 Sanity Checks
# data.frame(levels(vote_actual$justice), rownames(Vmatrix))      # justice list matches
# data.frame(levels(vote_actual$citation), rownames(combnet_adj_ho))  # holdout case names match

# 12.4 Set up testing vectors
N <- nrow(vote_actual)
pred_justice <- as.numeric(vote_actual$justice)
pred_case <- as.numeric(vote_actual$citation)

# 12.5 Set up STAN
predict_dat <- list (
          K = nrow(combnet_adj_tr),
          K_til = nrow(combnet_adj_ho),

          alpha_c = alpha_c,
          delta = delta,
          beta = beta,
          x = x,
          Citation = as.matrix(combnet_adj_ho),         # STAN doesn't recognize sparse matrix

          sigma_x1 = sigma_x1,
          sigma_x2 = sigma_x2,
          rho_x = rho_x,

          sigma_gamma = sigma_gamma,

          alpha_a = alpha_a,
          y = y,
          sigma_y = sigma_y,

          N = nrow(vote_actual),
          justice = pred_justice,
          casename = pred_case,
          J = nrow(Vmatrix),

          V = Vmatrix
          )

predict_fit <- stan(file = paste(filename, "_predict.stan", sep=""), init="random", data = predict_dat, iter = 1000, chains = 1)
save(predict_fit, cite_fit, agree_fit, procrustes.out, votedata_tr, votedata_ho, casedata_tr, casedata_ho, combnet_adj_tr, combnet_adj_ho, caselookup, file=paste(filename, ".Rdata", sep=""))
```

```
# 13. Calculate Predictions
predict_results <- get_posterior_mean(predict_fit)
p_a <- predict_results[grep("p_a\\[", rownames(predict_results)),]
p_r <- predict_results[grep("p_r\\[", rownames(predict_results)),]

prediction <- round(p_a / (p_a + p_r))
vote_actual$prediction <- prediction
vote_actual
table(vote_actual$affirmvote, vote_actual$prediction)
```

Additional Analysis

```
# Initialization
rm(list = ls())    # clear all

setwd("/Users/ekcheng/Documents/Dissertation/Chapter3/Code/")
load("latent7d.Rdata")

# Load data from end of latent7d
# Has Ruger data manually added
votecomp <- read.csv("RugerPred.csv")

# 30% Predictive accuracy of the model with SCJD outcomes
acc.model <- table(votecomp$affirmvote, votecomp$prediction)
(acc.model[1,1] + acc.model[2,2]) / sum(acc.model)

# Predictive accuracy of Ruger (82%)
# Have to remove Madigan v. Telemarketing Associates because it wasn't granted cert
    until after study started
votecomp2 <- votecomp[!is.na(votecomp$rugeractual),]
acc.rugerSCJD <- table(votecomp2$rugerpred, votecomp2$affirmvote)
(acc.rugerSCJD[1,1] + acc.rugerSCJD[2,2]) / sum(acc.rugerSCJD)

# Predictive accuracy of Ruger (80%)
# Have to remove Madigan v. Telemarketing Associates because it wasn't granted cert
    until after study started
votecomp2 <- votecomp[!is.na(votecomp$rugeractual),]
acc.rugerweb <- table(votecomp$rugerpred, votecomp$rugeractual)
(acc.rugerweb[1,1] + acc.rugerweb[2,2]) / sum(acc.rugerweb)

# 40% Predictive accuracy of the model with Ruger outcomes
acc.modelweb <- table(votecomp$rugeractual, votecomp$prediction)
(acc.modelweb[1,1] + acc.modelweb[2,2]) / sum(acc.modelweb)

# Calculate Predictions
predict_results <- get_posterior_mean(predict_fit)
p_a <- predict_results[grep("p_a\\[", rownames(predict_results)),]
p_r <- predict_results[grep("p_r\\[", rownames(predict_results)),]

prediction <- round(p_a / (p_a + p_r))
vote_actual$prediction <- prediction
vote_actual
table(vote_actual$affirmvote, vote_actual$prediction)
```

Stan Code

Citation Space Model

```
# STAN model file for Supreme Court latent space classifier
# Based on Eq. 1

data {
  int<lower=1> K;                    // number of cases
  int<lower=0, upper=1> Citation[K,K];     // citation matrix
  real alpha_a;                    // Agreement model output
  real y[K];                       // Latent ideology space
  }

parameters {
  real alpha_c;                  // intercept for citation model
  vector[2] x[K];                  // 2-D latent position for training cases
  real<lower=0,upper=100> sigma_x1;
    real<lower=0,upper=100> sigma_x2;
  real<lower=-1,upper=1> rho_x;
  real gamma[K];
  real delta[K];
  real<lower=0,upper=100> sigma_gamma;     // hyperparameter for random effs
  real<lower=0,upper=100> sigma_delta;
  real beta;
  }

model {
  vector[2] mu;
  matrix<lower=0>[2,2] Sigma_x;
  Sigma_x[1,1] <-  square(sigma_x1);
  Sigma_x[2,2] <- square(sigma_x2);
  Sigma_x[1,2] <- rho_x * sigma_x1 * sigma_x2;
  Sigma_x[2,1] <- Sigma_x[1,2];

  mu[1] <- 0;
  mu[2] <- 0;

// hyperpriors
  sigma_x1 ~ cauchy(0,5);        // Suggested in STAN manual p.147
  sigma_x2 ~ cauchy(0,5);
  rho_x ~ uniform(-1, 1);

  sigma_gamma ~ cauchy(0,5);
  sigma_delta ~ cauchy(0,5);

// priors
  for (i in 1:K) {
    x[i] ~ multi_normal(mu,Sigma_x);
    }

  alpha_c ~ normal(0,100);
  gamma ~ normal(0, sigma_gamma);
  delta ~ normal(0, sigma_delta);
  beta ~ normal(0,100);

// model
  for (k in 2:K) {
    for (l in 1:(k-1)) {
      Citation[k,l] ~ bernoulli_logit(
        alpha_c + gamma[k] + delta[l] - squared_distance(x[k], x[l])
        + beta * inv_logit(alpha_a - (y[k] - y[l])^2)
        );
      }
    }
}
```

Judge Agreement Model

```
data {

// Training
  int<lower=1> K;                  // number of cases
  int<lower=1> N_z;                // agreement observations
  int<lower=0, upper=1> AgreeZ[N_z];       // agree matrix
  int case1_z[N_z];                // case1 for agreement
  int case2_z[N_z];                // case2 for agreement
  }

parameters {
  real alpha_a;                // intercept for citation model
  real y[K];                   // 1-D latent position for ideology
  real<lower=0,upper=100> sigma_y;
  }

model {
// hyperpriors
  sigma_y ~ cauchy(0,5);

// priors
  alpha_a ~ normal(0,100);
  y ~ normal(0,sigma_y);

// model
  for (i in 1:N_z) {
    AgreeZ[i] ~ bernoulli_logit(
        alpha_a - (y[case1_z[i]] - y[case2_z[i]])^2
        );
    }
} // end of model section
```

Prediction Model

```
# STAN model file for Supreme Court latent space classifier
# Based on Eq. 1

data {

// Training
  int<lower=1> K;                  // number of training cases
  int<lower=1> K_til;                 // number of test cases

  real alpha_c;                  // citation model
  real delta[K];                   //
  real beta;
  vector[2] x[K];                  // latent coordinates
  int<lower=0, upper=1> Citation[K_til, K]; // citation

  real<lower=0> sigma_x1;
  real<lower=0> sigma_x2;
  real<lower=-1, upper=1> rho_x;

  real<lower=0> sigma_gamma;

  real alpha_a;                  // agreement model
  real y[K];                     //
  real<lower=0> sigma_y;

  int<lower=1> N;                  // Desired Predictions
  int<lower=1> justice[N];
  int<lower=1> casename[N];
  int<lower=1> J;

  real<lower=-1, upper=1> V[J, K];        //
```

```
  }

parameters {
  vector[2] x_ho[K_til];            // 2-D latent position for test cases
  real gamma[K_til];
  real y_ho[K_til];
  }


model {
//  declarations -- STAN requires variable declarations first
  vector[2] mu;
  matrix<lower=0>[2,2] Sigma_x;

  Sigma_x[1,1] <-  square(sigma_x1);
  Sigma_x[2,2] <- square(sigma_x2);
  Sigma_x[1,2] <- rho_x * sigma_x1 * sigma_x2;
  Sigma_x[2,1] <- Sigma_x[1,2];

  mu[1] <- 0;
  mu[2] <- 0;

// priors
  for (k_til in 1:K_til) {
    x_ho[k_til] ~ multi_normal(mu,Sigma_x);
    }

  gamma ~ normal(0, sigma_gamma);
  y_ho ~ normal(0,sigma_y);

// model
  for (k_til in 1:K_til)  {
    for (l in 1:K) {
      Citation[k_til,l] ~ bernoulli_logit(
        alpha_c + gamma[k_til] + delta[l] - squared_distance(x_ho[k_til], x[l])
        + beta * inv_logit(alpha_a - (y_ho[k_til] - y[l])^2)
        );
      }
    }

} // end of model section

generated quantities {
// declarations
  real<lower=0, upper=1> theta[K_til, K];
  real<lower=0, upper=1>  p_a[N];
  real<lower=0, upper=1>  p_r[N];

  for (k_til in 1:K_til)  {
    for (l in 1:K) {
      theta[k_til, l] <- inv_logit(alpha_a - (y_ho[k_til] - y[l])^2);
      }
    }

  for (i in 1:N) {
    p_a[i] <- 1;
    p_r[i] <- 1;

    for (l in 1:K) {
      if (V[justice[i],l] == 0) {
        p_a[i] <- p_a[i];
        p_r[i] <- p_r[i];
        }
      else {
        p_a[i] <- p_a[i] * (theta[casename[i], l] ^ (0.5 * (1 + V[justice[i],l])))
            * ((1 - theta[casename[i],l]) ^ (0.5 * (1 -  V[justice[i],l] )));

        p_r[i] <- p_r[i] * (theta[casename[i], l] ^ (0.5 * (1 - V[justice[i],l])))
```

```
                    * ((1 - theta[casename[i],l]) ^ (0.5 * (1 +  V[justice[i],l] )));
                }
            }
        }
}
```